

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS ARARANGUÁ
TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO

STELA RONCHI MENEGON

**SOFTGAME: JOGO DE TABULEIRO COMO MATERIAL DE APOIO À
ENGENHARIA DE SOFTWARE**

Araranguá, 13 de julho de 2016

STELA RONCHI MENEGON

SOFTGAME: JOGO DE TABULEIRO COMO MATERIAL DE APOIO À ENGENHARIA DE SOFTWARE

Trabalho de Curso submetido à Universidade Federal de Santa Catarina como parte dos requisitos necessários para a obtenção do Grau de Bacharel em Tecnologias da Informação e Comunicação. Sob a orientação da Professora Dra. Luciana Bolan Frigo.

Araranguá, 2016

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Menegon, Stela Ronchi

Softgame: jogo de tabuleiro como material de apoio à
engenharia de software / Stela Ronchi Menegon ;
orientadora, Luciana Bolan Frigo - Araranguá, SC, 2016.
119 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Campus Araranguá.
Graduação em Tecnologias da Informação e Comunicação.

Inclui referências

1. Tecnologias da Informação e Comunicação. 2. Engenharia
de Software. 3. Jogos Analógicos. 4. Jogos na Educação. 5.
Jogo de tabuleiro. I. Frigo, Luciana Bolan . II.
Universidade Federal de Santa Catarina. Graduação em
Tecnologias da Informação e Comunicação. III. Título.

Stela Ronchi Menegon

**SOFTGAME: JOGO DE TABULEIRO COMO MATERIAL DE APOIO À
ENGENHARIA DE SOFTWARE**

Trabalho de Curso submetido à
Universidade Federal de Santa Catarina
como parte dos requisitos necessários
para a obtenção do Grau de Bacharel em
Tecnologias da Informação e
Comunicação



Luciana Bolan Frigo
Doutora/UFSC



Eliane Pozzebon
Doutora/UFSC



Adriano Oliveira
Professor/UFSC

Araranguá, 13 de julho 2016

*Dedico este trabalho a minha família e
amigos, a todos que, de alguma maneira, me
ajudaram a chegar até aqui.*

AGRADECIMENTOS

Agradeço a minha família pelo incentivo, apoio e estímulo para enfrentar as barreiras da vida.

Agradeço a minha querida amiga Heloisa, pela paciência e parceria em todos os momentos, que além de ser a melhor designer é minha melhor amiga e me apoiou na realização deste trabalho.

Agradeço à minha orientadora, Luciana, por ser uma excelente professora e profissional que me ajudou nesta etapa final.

Agradeço ao meu professor Adriano, por ter disponibilizado sua aula para aplicar o jogo aos alunos, e por ter me dado todo apoio para a realização do projeto.

E finalmente agradeço a Deus, por proporcionar estes agradecimentos à todos que tornaram minha vida mais afetuosa, além de ter me dado uma família maravilhosa e amigos sinceros.

*A mente que se abre a uma nova ideia jamais
volta ao seu tamanho original.*
(Albert Einstein)

RESUMO

Jogos educacionais podem ser utilizados como uma ferramenta para auxiliar no processo de ensino-aprendizagem estimulando os alunos no desenvolvimento de habilidades físicas e mentais. Tais jogos fornecem uma maneira de colocar em prática conhecimentos obtidos em áreas que envolvem muitos conceitos, como é o caso da engenharia de software.

Dessa maneira, considerando a engenharia de software como uma disciplina consideravelmente teórica, mas de extrema importância para os cursos de computação, este trabalho propõe o desenvolvimento de um jogo de tabuleiro, que auxilie no processo de aprendizagem da engenharia de software, estimulando o aluno dentro da sala de aula.

O jogo traz uma abordagem teórica sobre conceitos e metodologias de engenharia de software, com perguntas classificadas em três níveis de complexidade divididas em quatro tópicos: introdução à engenharia de software, engenharia de requisitos, produção de software e modelos de processos, planejamento e gerência de projetos e qualidade, incluindo testes, manutenção e evolução de software. Este jogo de tabuleiro possui quatro peões, uma ampulheta, um dado personalizado, uma embalagem, um tabuleiro bem como suas regras.

Os resultados obtidos com o jogo SoftGame foram satisfatórios. Ele é um jogo descomplicado de ser jogado possuindo regras simples. A interação experimentada entre os jogadores, organizados em equipes, é um fator que estimula a diversão e a competição saudável entre os mesmos. Através da abordagem de assuntos referentes à engenharia de software, durante a partida os jogadores são estimulados a desafiar seu conhecimento para conseguir realizar as tarefas presentes neste jogo e saírem vencedores.

Palavras chaves: Jogo de tabuleiro, Engenharia de software, Jogos na Educação, Jogos Analógicos.

ABSTRACT

Educational games can be used as a tool to assist in the teaching-learning process encouraging students to develop physical and mental skills. Such kits provide a way of putting into practice knowledge obtained in areas involving many concepts, as is the case of software engineering.

Thus, considering the software engineering as a pretty theoretical discipline, but very important for computer courses, this work proposes the development of a board game that helps the software engineering learning process, encouraging the student in the classroom.

The game brings a theoretical approach to concepts and software engineering methodologies, with questions classified into three levels of complexity divided into four topics: introduction to software engineering, requirements engineering, software production and process models, planning and management designs and quality, including testing, maintenance and evolution. This board game has four pawns, an hourglass, a custom data, a package, a board and its rules.

The results obtained with the game SoftGame were satisfactory. It is an uncomplicated game, because it has simple rules. The interaction between the players organized into teams, is the factor that encourages fun and healthy competition between them. By addressing issues related to software engineering, during the game players are encouraged to challenge their knowledge to accomplish the tasks present in this game and come out winners.

Key words: Board Game, Software Engineering, Games in Education, Analogical Games.

LISTA DE ILUSTRAÇÕES

Figura 1 - Jogo de tabuleiro SimuleS	34
Figura 2 - Jogo SimSE.....	35
Figura 3 - MO-SEProcess – Ambiente de Jogo	36
Figura 4 - Jogo de tabuleiro PM Master.....	37
Figura 5 - Tela do jogo “A Ilha dos Requisitos”	37
Figura 6 - Jogo Lego SCRUM City.....	38
Figura 7 -Tabuleiro do Jogo AprendES	39
Figura 8 - Tabuleiro do Jogo SoftGame.....	43
Figura 9 - Frente Carta	45
Figura 10 - Verso Carta	45
Figura 11 - Frente Carta Hacker.....	46
Figura 12 - Verso Carta Hacker	46
Figura 13 – Alunos jogando SoftGame.....	50
Figura 14 – Avaliação Nível Motivação	55
Figura 15– Avaliação Nível Experiência do Usuário.....	56

Figura 16 – Avaliação Nível Aprendizagem.....	56
---	----

LISTA DE QUADROS

Quadro 1 - Questões afirmativas para avaliação de jogos educacionais.....	51
---	----

LISTA DE ABREVIATURAS E SIGLAS

ES - Engenharia de Software

UML - Linguagem de Modelagem Unificada

IES - Instituição de Ensino Superior

PMBOK - Project Management Body of Knowledge

SUMÁRIO

1. INTRODUÇÃO.....	15
1.1. Problema.....	16
1.2. Objetivos gerais	16
1.3. Objetivos específicos.....	16
1.4. Justificativa.....	16
1.5. Metodologia.....	17
1.6. Organização do Texto	18
2. JOGOS ANALÓGICOS.....	19
2.1. Jogos de Tabuleiro	21
2.2. Interação Lúdica	22
2.3. Jogos e suas Regras	25
2.3.1. Três tipos de regras	26
3. JOGOS NA EDUCAÇÃO	28
3.1. Ensino de Engenharia de Software	30
3.2. Jogos Educacionais para Engenharia de Software.....	33
4. ENGENHARIA DE SOFTWARE, UMA VISÃO GERAL	40
5. VISÃO GERAL DO JOGO	42
5.1. Objetivo do jogo	42
5.2. Componentes do jogo	42
6. DINAMICA E PREPARAÇÃO DO JOGO	47
6.1. O Jogo.....	47
6.1.1. O juiz	48
6.1.2. O Vencedor	48
6.2. Conteúdo do Jogo	48
7. RESULTADOS OBTIDOS.....	50
7.1. Análise do questionário aplicado.....	54
8. CONSIDERAÇÕES FINAIS	57
REFERÊNCIAS	58
ANEXOS	61
APÊNDICE	63

INTRODUÇÃO

A criação de jogos para auxiliar na aprendizagem é uma tarefa desafiadora para os instrutores que não possuem disponibilidade ou conhecimento suficiente para criar seus jogos de forma efetiva.

Os alunos de computação precisam adquirir competências em diversas áreas, incluindo algoritmos, programação, engenharia de software, redes de computadores, entre outras, ACM/IEEE (2005 apud WANGENHEIM, 2012, p.13) e isso em diversos níveis de aprendizagem, incluindo o nível de conhecimento em que o aluno aprende a lembrar dos conceitos, o nível de compreensão em que ele aprende a classificar, estruturar e organizar o conhecimento e o nível de aplicação em que o aluno se torna capaz de aplicar o conhecimento em situações concretas. Eles ainda precisam aprender não só o conhecimento, mas também habilidades como trabalhar em equipe e desenvolver atitudes profissionais (WANGENHEIM, 2012, p. 13).

No entanto, a tradicional forma de ensino, extremamente centrada no professor, faz com que falte aos estudantes oportunidades para aplicação prática dos conceitos aprendidos em aula.

Recentemente, criou-se a expectativa de que jogos e dinâmicas educacionais sejam um meio bastante vantajoso para completar o ensino das competências relacionadas à área de computação. (WANGENHEIM, 2012, p. 16)

Dessa forma, o presente trabalho tem como objetivo a criação de um jogo de tabuleiro para ser usado como material de apoio em sala de aula na disciplina de Engenharia de Software.

O Jogo de tabuleiro é um meio de interação entre pessoas sem dúvida alguma e ferramenta de apoio para o ensino da computação. Atualmente existem diversos jogos de ta-

buleiro no mercado e todos por mais diferentes que sejam, apresentam um objetivo em comum, fazer os jogadores se divertirem e alcançarem um objetivo em comum que é vencer.

As pesquisas realizadas foram retiradas de livros, artigos, páginas da Web e jogos de tabuleiro já produzidos, e englobam os principais conceitos da Engenharia de Software, divididos em quatro tópicos, facilitando o entendimento da disciplina e ao mesmo tempo motivando os alunos e fomentando o interesse a esta área.

1.1.Problema

Como criar um jogo de tabuleiro que proponha interação entre os alunos e os motive na busca de conceitos da engenharia de software?

1.2.Objetivos gerais

Desenvolver um jogo de tabuleiro como material de apoio para o ensino e aprendizagem da engenharia de software.

1.3.Objetivos específicos

Para que o objetivo geral seja alcançado, será necessário o cumprimento das seguintes etapas:

1. Fundamentar teoricamente o tema da pesquisa;
2. Pesquisar os conteúdos mais relevantes na aprendizagem da engenharia de software;
3. Desenvolver um jogo de tabuleiro voltado para estudantes de engenharia de software;
4. Testar o jogo e avaliar qualitativamente.

1.4. Justificativa

A aprendizagem lúdica através do jogo está se tornando um fator diferencial para o ensino.

A interação do jogo com a aprendizagem lúdica desenvolve a criatividade, a sociabilidade e as inteligências múltiplas, que são fatores essenciais para os estudantes da área da computação.

No processo de ensino, quanto mais ideias melhor, pois a disciplina de Engenharia de software engloba muita teoria, o que acaba tornando as aulas cansativas e desestimulantes.

Wangenheim (2012, p.15) afirma que aulas expositivas são consideradas chatas e não ajudam em motivar o aluno a aprender, levando em consideração que os alunos de hoje representam a geração Y, já acostumados com o comportamento intensamente interativo e esperam sempre um retorno imediato.

Em virtude da necessidade de capacitar o aluno nos processos da Engenharia de Software, uma das alternativas é a utilização de jogos para preencher a lacuna existente entre o teórico e o prático, e também para suprir a necessidade dessa interação que existe nos novos alunos.

A importância de um jogo com foco na Engenharia de Software vem para suprir essa forma de desenvolvimento intelectual de maneira mais prática, que o acadêmico de fato seja estimulado brincando.

Justifica-se que existe uma necessidade de desenvolver um game com foco na Engenharia de Software, para que os acadêmicos e profissionais da área desenvolvam ainda mais seus conhecimentos e estimulem sua criatividade.

1.5. Metodologia

O presente trabalho tem como foco principal a construção de um jogo de tabuleiro para auxílio no ensino aprendizagem da Engenharia de Software.

Na primeira etapa do trabalho é realizada uma pesquisa bibliográfica de conceitos de Jogos, Jogos de tabuleiro, Interação Lúdica, conceitos sobre as principais regras de um jogo fornecendo, desta forma, a base sobre termos e conceitos que são utilizados ao longo do trabalho.

Na segunda etapa do trabalho é realizado um estudo sobre os jogos na educação, sobre os jogos já existentes na área da Engenharia de software para fornecer conhecimento na

continuidade do trabalho, nesta etapa inclui também uma breve pesquisa sobre engenharia de software.

Na terceira etapa está o desenvolvimento do protótipo: definição das regras, dos elementos do jogo, design do tabuleiro, das cartas e do dado, como também todo o conteúdo da engenharia de Software usado na construção do jogo.

A quarta e última etapa é a realização dos testes, o jogo é testado em sala de aula por alunos que cursam a disciplina de engenharia de software, o jogo é avaliado qualitativamente através de questionário aplicado aos alunos depois da experiência do jogo.

1.6. Organização do Texto

Este trabalho está estruturado em seis capítulos. Este primeiro capítulo apresenta a descrição do problema, objetivos gerais e específicos, a justificativa e as metodologias de pesquisa adotadas.

O segundo capítulo contém um estudo sobre a história dos jogos e jogos de tabuleiro, interação lúdica e também sobre as regras do jogo.

O terceiro capítulo apresenta um estudo sobre os jogos educativos, bem como os jogos voltados ao ensino de Engenharia de Software.

No quarto capítulo é apresentada uma visão geral sobre o que é Engenharia de Software.

O quinto capítulo explica o jogo SoftGame, os objetivos, principais elementos, dinâmica e preparação, e também o conteúdo da Engenharia de Software selecionada para a confecção do jogo, além de apresentar a experiência do uso do jogo em sala de aula.

O sexto capítulo apresenta as considerações finais sobre o desenvolvimento e aplicação do jogo.

2. JOGOS ANALÓGICOS

A origem dos jogos analógicos é considerada milenar, registros de jogos de tabuleiro foram encontrados em todo o mundo tem sua origem desconhecida, pois somente após um longo período de práticas populares, que descendem do cotidiano lúdico dos povos, eles começaram a ser estudados.

Devido a estas práticas terem força e expressão considerável, sua transmissão entre as gerações ainda persiste, o que lhes compete o caráter de universalidade.

Segundo Santos (2006, p. 35)

“Atualmente, percebe-se em todo o mundo uma preocupação com o resgate do brincar, enquanto patrimônio lúdico-cultural da humanidade. Ao lado dessa tendência, também se percebe uma crescente valorização das brincadeiras tradicionais como forma de conhecimento dos grupos sociais e das diversidades culturais dos povos.”

Para estimular a consciência cultural das pessoas, como também resgatar os valores através da utilização dos jogos tradicionais que unem educação, cultura e ludicidade, a educação seria o caminho mais eficiente.

De acordo com Leal (2005, p. 112) o jogo é uma atividade lúdica em que crianças ou adultos participam de uma situação de engajamento social num tempo e espaços determinados, como características próprias delimitadas pelas próprias regras de participação na situação “imaginária”.

Foram delimitados dois grupos principais de jogos: jogos de enredo e jogos de regras. Os jogos de enredo têm recebido várias denominações: jogo imaginativo, jogo de faz de conta, jogo de papéis, jogo simbólico ou jogo sócio-dramático. “A ênfase é dada à simulação ou faz de conta, cuja importância é ressaltada por pesquisas que mostram sua eficácia para

promover o desenvolvimento cognitivo e afetivo-social da criança” (BOMTEMPO, 2003, p. 58).

Macedo (1995, p. 7) defende que:

Os jogos simbólicos caracterizam-se pela assimilação deformante (PIAGET, 1945). Deformante porque nessa situação a realidade (social, física etc.) é assimilada por analogia, como a criança pode ou deseja, isso é os significados que ela dá para os conteúdos de suas ações, quando joga, são deformações – maiores ou não – dos significados correspondentes na vida social ou física. Graças a isso, pode compreender as coisas, afetiva ou cognitivamente, segundo os limites de seu sistema cognitivo.

Moura (2003, p. 79-80), também defendendo essa ideia, argumenta que:

Nesta concepção, o jogo promove o desenvolvimento, porque está impregnado de aprendizagem. E isto ocorre porque os sujeitos, ao jogar, passam a lidar com regras que lhes permitem a compreensão do conjunto de conhecimentos veiculados socialmente, permitindo-lhes novos elementos para apreender os conhecimentos futuros.

O outro tipo de jogo é o jogo de regras. Nesse, a situação imaginária está implícita, e as regras orientam a brincadeira. Os participantes do jogo centram a atenção na finalidade do jogo e no atendimento às regras compartilhadas. A situação imaginária fica oculta. O que caracteriza o jogo de regras é: 1) existência de um conjunto de leis imposto pelo grupo, sendo que seu descumprimento é normalmente penalizado, e 2) existência da forte competição entre os indivíduos. Ao jogar xadrez, por exemplo, a atenção volta-se para as regras, e não para o exercício de praticar o papel de rainha ou rei, mas a situação imaginária é que orienta os tipos de regra. (MORAIS et al., 2005, p. 114).

Atualmente, a mídia, vem estimulando a compra de jogos de computadores, e que em consequência estes jogos tradicionais acabam se perdendo no tempo.

E isso é uma questão a ser considerada, rever os conceitos sobre o lúdico, e a sua importância para o desenvolvimento de uma gama de possibilidades de desenvolvimento e aprendizagem, além de que esses jogos possuem um papel importante na preservação da cultura.

2.1. Jogos de Tabuleiro

Um jogo é uma prática estimulante e lúdica que pode envolver qualquer tipo de competição com regras criadas num ambiente restrito ou até mesmo imediato, diferente do esporte, cujas regras são universais. (ZUCARELLI; COUTO, 2013, p. 24).

O jogo de tabuleiro envolve contadores ou peças movidas ou colocadas sobre uma superfície pré-demarcada que segue uma ordem ou “board”, seguindo um conjunto de regras. “Os jogos podem basear-se em estratégia, azar (por exemplo, rolando dados) ou uma mistura dos dois, e geralmente têm um objetivo que o jogador pretende alcançar.” (ZUCARELLI; COUTO, 2013, p. 24).

Zucarelli e Couto (2013, p. 24) também afirma que existem muitos estilos de jogos de tabuleiro, podendo variar de situações da vida real a jogos abstratos sem nenhum tema inerente, como a damas ou uma narrativa específica. Podem ter regras ou descreverem um universo com grandes detalhes.

Os jogos de tabuleiros têm como características a socialização, segundo Santos (2013), o jogo é social quando estimula os alunos a se relacionarem entre si durante as partidas, bem como os incentiva a obedecerem às regras e limites do adversário. A área afetiva ocorre no respeito com a vez do colega, durante a partida, bem como no “saber ganhar e no saber perder”, compreendendo que esta prática é inerente ao jogo, e que aquele que ganha, não é melhor do que aquele que perde.

O lado cognitivo diz respeito às competências acadêmicas desenvolvidas pelo estudante com as jogadas, como por exemplo: habilidades de raciocínio, estratégia, comunicação, administração, inteligência emocional, liderança, concentração, negociação, entre outras. (AMBRÓSIO; SCHOEFFEL, 2014, p.3)

Jogos de tabuleiro foram jogados na maioria das culturas e sociedades ao longo da história, (ZUCARELLI; COUTO, 2013, p. 25) divide em cinco categorias:

- Jogos Antigos: São jogos introduzidos ao longo da história da humanidade antes da Revolução Industrial.

- Jogos de Mercado de Massa: São os jogos lançados pelas grandes companhias de brinquedos pelo mundo todo, grande parte dos jogos que todos conhecemos se enquadra nessa categoria.
- Jogos de Guerra (Wargames): Jogos que simulam um conflito armado, real ou imaginário.
- Jogos de Estilo Americano: São jogos onde o tema do jogo é mais importante que o jogo em si, geralmente com várias regras para suportar o tema.
- Jogos de Estilo Europeu: São jogos onde a jogabilidade é mais importante do que o tema, sendo este, muitas vezes inexistente.

A prática dos jogos de tabuleiro incentiva a capacidade de memória, ajudam a desenvolver o raciocínio lógico e abstrato, além de ser uma alternativa de lazer.

2.2. Interação Lúdica

Johann Huizinga (2014) é considerado um dos maiores estudiosos de jogos do século 20. *Homo Ludens*, seu livro, descreve o papel do jogo na civilização humana.

Huizinga (2014, p.4) afirma que brincadeiras e jogos, que têm sido difamados na história recente como triviais e fúteis, estão na verdade, bem no centro do que nos torna humanos. Ele afirma que “o jogo é mais antigo que a cultura”.

[...]o jogo é mais do que um fenômeno fisiológico ou um reflexo psicológico. Ultrapassa os limites da atividade puramente física ou biológica. É uma função significante, isto é, encerra um determinado sentido. No jogo existe alguma coisa “em jogo” que transcende as necessidades imediatas da vida e confere um sentido à ação. Todo Jogo significa alguma coisa. (HUIZINGA, 2014, p.3-4)

Neste trecho do livro o autor enfatiza o fato que todo jogo tem um significado, que existe um “sentido” para o jogo.

Há uma extraordinária divergência entre as numerosas tentativas de definição da função biológica do jogo. Umas definem as origens e o fundamento do jogo em termos de descarga de energia vital superabundante, outras, como satisfação de um certo “instinto de imitação”, ou ainda, simplesmente como uma “necessidade” de distensão. Segundo uma teoria, o jogo constitui uma preparação do jovem para as tarefas sérias que mais tarde a vida dele exigirá; segundo outra, trata-se de um exercício

de autocontrole indispensável ao indivíduo. Outras veem o princípio do jogo como um impulso inato para exercer certa faculdade, ou como o desejo de dominar ou competir. Teorias há, ainda, que o consideram um “ab-reação” um escape para impulsos prejudiciais, um restaurador de energia aprendida por uma atividade unilateral, ou “realização do desejo”, ou uma ficção destinada a preservar o sentimento do valor pessoal etc. (HUIZINGA, 2014, p.4)

Compreende através desta citação que naquela época já não faltavam teorias que tentassem de alguma maneira explicar o ato de jogar. No entanto, elas subentendiam que o jogo estivesse ligado a alguma finalidade biológica, a um aprendizado, e não se preocupavam em saber o que é o jogo em si ou o que ele significa para os jogadores.

Huizinga (2014, p. 16) define a natureza e o significado do ato lúdico, aquilo que em português descrevemos como brincar, encenar, jogar.

Numa tentativa de resumir as características formais do jogo, poderíamos considerá-lo uma atividade livre, conscientemente tomada como “não-séria” e exterior à vida habitual, mas ao mesmo tempo capaz de absorver o jogador de maneira intensa e total. É uma atividade desligada de todo e qualquer interesse material, com a qual não se pode obter qualquer lucro, práticas dentro de limites espaciais e temporais próprios, segundo uma certa ordem e certas regras. Promove a formação de grupos sociais com tendências a rodearem-se de segredos e a sublinharem sua diferença em relação ao resto do mundo por meio de disfarces ou outros meios semelhantes. (HUIZINGA, 2014, p.16)

Huizinga (2014, p. 16) define a função do jogo pelos dois aspectos fundamentais encontrados no próprio jogo “uma luta por alguma coisa ou a representação de alguma coisa”.

A partir dessa definição Santanella e Feitoza (2009, p.6) concluem em seu livro Mapa do jogo: a diversidade cultural dos games, que: 1) um jogo é uma atividade livre; 2) o jogador tem consciência de que o jogo é uma atividade exterior à vida habitual; 3) o jogo não tem finalidade lucrativa; 4) acontece dentro de um espaço e tempo definidos; 5) de todo jogador se espera imersão e seriedade na execução da atividade, ou seja, aquele que não joga seriamente é tido com uma desmancha-prazeres; 6) o jogo é uma atividade que agrega clãs que compartilham interesses; 7) todo jogo tem objetivos, metas, finalidades; 8) todo jogo tem regras e certa ordem.

Rogers Caillois em seu livro “Os Jogos e os homens” discorda em alguns pontos do pensamento de Huizinga, Caillois não concordando com o desinteresse material que é defendido por Huizinga.

Para Caillois (1990, p. 5-6) “o jogo é ocasião de puro gasto: de tempo, de energia, de perspicácia, de destreza e, muitas vezes, de dinheiro para adquirir o equipamento do jogo ou eventualmente pagar para um estabelecimento [onde se joga]”.

Segundo Salen e Zimmerman (2012, p. 49) uma das dificuldades na identificação da interação lúdica significativa nos jogos é a variedade quase infinita de forma que o jogo pode assumir. Os autores citam alguns exemplos, como:

- O duelo intelectual em uma partida de xadrez.
- O balé e a improvisação em equipe do basquete.
- A mudança dinâmica das identidades individuais e comunitárias nos papéis on-line no jogo EverQuest.
- O invasivo jogo Assassin, jogado em um campus universitário.

Mas o que todos esses exemplos têm em comum? O que os autores querem dizer com isso?

Para Salen e Zimmerman (2012, p.49) cada um situa a ação no contexto de um jogo. “A ação não vem apenas do jogo em si, mas da maneira como os jogadores interagem com o jogo para jogá-lo”.

Em outras palavras, o tabuleiro, as peças e até mesmo as regras do xadrez não podem sozinhos criar uma interação lúdica significativa. A interação lúdica significativa surge da interação entre jogadores e os sistema do jogo, bem como do contexto em que o jogo é jogado. Uma forma de estabelecer o que os jogadores fazem quando jogam um jogo é dizer que eles estão fazendo escolhas. (SALEN; ZIMMERMAN, 2012, p.49).

Os jogadores estão decidindo como mover suas peças, como mover seus corpos, quais cartas jogar, quais opções escolherem, quais estratégias adotarem, como interagir com outros jogadores. “Eles ainda têm de fazer a escolha se devem ou não jogar!” (SALEN; ZIMMERMAN, 2012, p.49).

Quando um jogador faz uma escolha em um jogo, a ação que resulta da escolha tem um desfecho. No xadrez, se um jogador move uma peça no tabuleiro, essa ação afeta

as relações de todas as outras peças: uma peça pode ser capturada ou um rei pode de repente encontrar-se em xeque. Em, *Assassin*, se um jogador persegue furtivamente seu alvo e consegue atirar nele com uma pistola de dardos, o jogo todo muda como resultado dessa ação: uma pontuação é marcada, a vítima está fora no resto do jogo e deve dar o nome do alvo dela para o jogador que acabou de eliminá-la. Em *EverQuest*, se você mobilizar e matar um monstro, as estatísticas e o equipamento do seu personagem poderão mudar; o mundo-jogo maior também é afetado, mesmo que isso signifique simplesmente que, no momento, há menos um monstro. (SALEN; ZIMMERMAN, 2012, p.49).

Salen e Zimmerman (2012, p.49) afirmam que: “Jogar um jogo significa fazer escolhas e tomar medidas.” E toda essa atividade acontece dentro de um sistema de jogo desenhado para apoiar tipos significativos de escolhas, sendo que cada ação resulta em uma mudança que afeta o sistema global do jogo.

Outra forma de dizer isto seguindo ainda o raciocínio de Salen e Zimmerman é que uma ação que um jogador toma em um jogo resulta na criação de novos significados no sistema. Por exemplo, depois de mover uma peça do xadrez, as relações recém estabelecidas entre as peças do xadrez dão origem a um novo conjunto de significados – significados criados pela ação do jogador. (SALEN; ZIMMERMAN, 2012, p.49)

2.3. Jogos e suas Regras

Jogos têm regras, e jogar um jogo é seguir suas regras, elas constituem a estrutura formal interna dos jogos. E esta talvez seja a característica mais importante dos jogos, a que os distingue dos outros meios de comunicação, arte e conhecimento.

Para Huizinga (2014, p.33), dentro do conceito de jogo existem alguns aspectos a ser considerados, tais como a voluntariedade, a limitação de tempo e espaço para sua realização, o consentimento, porém obrigatoriedade das regras, além do sentimento de alegria que ele proporciona e a consciência de ser diferente da ‘vida cotidiana’.

Nota-se a importância que Huizinga (2014, p.33) confere às regras dentro do jogo, ao conceituá-lo como uma atividade voluntária que ocorre em tempo e espaço definidos e segundo ‘regras livremente consentidas, mas absolutamente obrigatórias’. A irresistível presença das regras no ambiente do jogo é também enfatizada pelo autor ao dizer:

(...) estas regras são um fator muito importante para o conceito de jogo. Todo jogo tem suas regras. São estas que determinam aquilo que ‘vale’ dentro do mundo temporário por ele circunscrito. As regras de todos os jogos são absolutas e não permitem discussão. (HUIZINGA, 2014, p.14).

Salen e Zimmerman (2012, p. 21-29) afirmam que “as regras não são a experiência do jogar”. É possível alterar, fazer alguns experimentos em um jogo, como por exemplo, alterar os nomes dos naipes de um baralho, sem que as regras ou estruturas formais sofram alterações.

As regras são o que diferenciam os jogos de outros tipos de brincadeira. Provavelmente a definição mais simples de um jogo seja a de que ele é uma brincadeira organizada, ou seja, baseada em regras. Se você não tem regras tão importantes para os jogos? Regras impõem limites - elas nos obrigam a tomar caminhos específicos para atingir metas e garantir que todos os jogadores assumam os mesmos caminhos. Elas nos colocam dentro do mundo do jogo, deixando-nos saber o que está dentro e fora dos limites. Marc Prensky (2001 apud SALEN; ZIMMERMAN, 2012, p. 26).

“As regras do jogo são diferentes das regras de etiqueta, leis, guerra ou outras regras sociais. Os jogos são intrinsecamente artificiais e separados de contextos “reais”, ao passo que essas outras formas de regras não estão separadas da vida comum.” (SALEN; ZIMMERMAN, 2012 p. 29).

Salen e Zimmerman (2012 p.26-28) consideram a seguinte lista de características da regra:

- As regras limitam a ação do jogador. Limitar as atividades dos jogadores é a principal maneira como as regras operam.
- As regras são explícitas e inequívocas. As regras são completas e não desfruta de qualquer ambiguidade.
- As regras são compartilhadas por todos os jogadores. Em um jogo com vários jogadores, todos obrigatoriamente compartilham um mesmo conjunto de regras.
- As regras são fixas. As regras de jogos são fixas e não se modificam quando um jogo é jogado.
- Regras são obrigatórias. As regras são feitas para serem cumpridas.
- Regras são repetíveis. As regras são repetíveis de jogo para jogo.

2.3.1. Três tipos de regras

Salen e Zimmerman (2012, p.33-34) propõem um sistema de três partes para entender o que são as regras do jogo e qual seu funcionamento.

- Regras Operacionais

As regras operacionais são as “regras do jogo”. Elas são consideradas como regras básicas do jogo, como as orientações de que os jogadores precisam para jogar.

- Regras Constitutivas

As regras constitutivas de um jogo são as estruturas formais lógicas e matemáticas subjacentes que existe “sob a superfície” das regras apresentadas aos jogadores.

- Regras Implícitas

Regras implícitas são as regras não escritas de um jogo. São as regras implícitas de comportamento apropriado do jogo, e que podem variar de jogo pra jogo.

3. JOGOS NA EDUCAÇÃO

“O jogo, como promotor da aprendizagem e do desenvolvimento, passa a ser considerado nas práticas escolares como importante aliado para o ensino” (MOURA, 2003, p. 80). No entanto, precisamos atentar que esses não podem ser utilizados como únicas estratégias didáticas nem garantem a apropriação dos conhecimentos que buscamos. Kishimoto (2003, p. 37-38) também colabora com essa posição quando diz que:

A utilização do jogo potencializa a exploração e a construção do conhecimento, por contar com a motivação interna, típica do lúdico, mas o trabalho pedagógico requer a oferta de estímulos externos e a influência de parceiros bem como a sistematização de conceitos em outras situações que não jogos.

Jogos educacionais tem a responsabilidade de favorecer e estimular a aprendizagem de crianças, jovens e adultos, através de um processo de socialização e coletividade que contribui para a formação de sua personalidade. Visando a estimulação do impulso natural do indivíduo a aprender.

Machado (2010, p. 18) afirma que:

Os jogos são ferramentas de instrução eficientes, usados como apoio à educação, principalmente pelo seu potencial motivacional, porque eles divertem enquanto motivam, facilitam o aprendizado e aumentam a capacidade de absorção do conteúdo a ser ensinado, estimulando as funções mentais e intelectuais do jogador, permitindo o reconhecimento e entendimento de regras, além de serem interativos.

Por serem dinâmicos e atraírem a atenção dos alunos os jogos educacionais tem uma grande importância na busca do conhecimento, pois colaboram como auxiliares no

ensino-aprendizagem de alunos que eventualmente perdem a concentração de suas atividades com métodos convencionais de ensino.

Tarouco et al. (2004, p.4), afirma que:

A utilização desses novos recursos modifica a dinâmica do ensino, as estratégias e o comprometimento de alunos e professores. Com esses novos recursos e ferramentas a educação pode ensinar uma aprendizagem significativa, proporcionando que o aluno aprenda de forma dinâmica e motivadora. Os avanços das tecnologias de softwares e linguagens de autoria facilitaram o processo de construção de jogos educacionais, fazendo com que professores possam passar de meros expectadores e avaliadores para produtores de jogos educacionais, capacitando-se para isso e podendo produzir softwares de qualidade, contextualizados com a sua realidade.

De uma forma geral, os jogos fazem parte da nossa vida desde os tempos mais antigos, estando presentes não só na infância, mas como em outros momentos. Os jogos podem ser ferramentas instrucionais eficientes, pois eles divertem enquanto motivam, facilitam o aprendizado e aumentam a capacidade de retenção do que foi ensinado, exercitando as funções mentais e intelectuais do jogador. (TAROUCO et al., 2004, p.1-2)

O jogo pode ser considerado como um importante meio educacional, pois propicia um desenvolvimento integral e dinâmico nas áreas cognitiva, afetiva, linguística, social, moral e motora, além de contribuir para a construção da autonomia, criatividade, responsabilidade e cooperação das crianças e adolescentes. (MORATORI, 2003, p. 9)

Além disso, também permitem o reconhecimento e entendimento de regras, identificação dos contextos que elas estão sendo utilizadas e invenção de novos contextos para a modificação das mesmas. Jogar é participar do mundo de faz de conta, dispor-se às incertezas e enfrentar desafios em busca de entretenimento. Através do jogo se revelam a autonomia, criatividade, originalidade e a possibilidade de simular e experimentar situações perigosas e proibidas no nosso cotidiano. (TAROUCO et al., 2004, p.2).

[...] os jogos podem ser empregados em uma variedade de propósitos dentro do contexto de aprendizado. Um dos usos básicos e muito importantes é a possibilidade de construir-se a autoconfiança. Outro é o incremento da motivação. [...] um método eficaz que possibilita uma prática significativa daquilo que está sendo aprendido. Até mesmo o mais simplório dos jogos pode ser empregado para proporcionar informa-

ções factuais e praticar habilidades, conferindo destreza e competência (SILVEIRA, 1998, p.2).

”Quando motivadores do processo de aprendizagem, eles podem ser definidos como jogos educacionais.” (TAROUCO et al.,2004, p.2)

Um jogo, para ser útil no processo educacional, deve promover situações interessantes e desafiadoras para a resolução de problemas, permitindo aos aprendizes uma auto avaliação quanto aos seus desempenhos, além de fazer com que todos os jogadores participem ativamente de todas as etapas. (MORATORI, 2003, p. 9)

De acordo com Soares (2012, p. 55) os jogos sérios desenvolvem características necessárias ao trabalho, pois pode ser utilizado como forma simular diversos elementos da realidade, como dirigir carros e realizar cirurgias. Os jogos educacionais são apenas instrumentos, por isso eles só serão úteis se acompanhados por alguém que analise o jogo e o jogador, de modo zeloso e crítico, que ao ver que tal ferramenta deixou de ser instrutiva e se transformou apenas numa disputa divertida, consiga sutilmente desenvolver um caminho certo ao aprendiz. Então, o papel do professor não deve ser o de guiar explicitamente os passos do aprendiz, mas sim não permitir que use este jogo sem entender nem aprender nada, e não permitir que o aprendiz se desvie muito do objetivo educacional (SOARES, 2012, p. 55).

Os jogos educacionais podem se transformar em uma alternativa importante e motivadora no processo de ensino-aprendizagem do aluno independente da idade.

3.1. Ensino de Engenharia de Software

Atualmente a sociedade em que vivemos exige uma formação e capacitação qualificada de profissionais, formar bons profissionais já se tornou parte do comprometimento das Instituições de Ensino Superior (IES), seja em cursos de graduação, pós-graduação ou de curta duração.

No ensino de Engenharia de Software (ES), especificamente, a qualidade dos profissionais está diretamente relacionada à qualidade da educação, embora também existam outros fatores que contribuem para isto. Beckman et al, (1997 apud PRIKLADNICKI et al., 2009, p.1)

Hoje se anuncia cada vez mais significativo tratar aspectos didáticos e pedagógicos no ensino. Aulas tradicionais, estratégia de ensino predominante, raramente satisfazem a necessidade de aprendizagem. Grillo (2002 apud PRIKLADNICKI et al., 2009, p.1).

Atualmente jogos educacionais são apontados como um tipo promissor de material instrucional, que pode trazer benefícios para a educação também na área de Engenharia de Software. Há indicações de que esse método de ensino costuma agradar e motivar os alunos, despertar o interesse e a curiosidade, além de contribuir positivamente na aprendizagem. (WANGENHEIM et al., 2009, p.4)

“A engenharia de software é uma disciplina preocupada com a teoria, conhecimento e práticas para a construção eficaz e eficiente de sistemas de software que satisfaçam as exigências dos usuários e clientes.” (ACM/IEEE, 2008, p.97, tradução nossa).

Um dos grandes desafios enfrentados no ensino da Engenharia de Software (ES) atualmente é suprir a necessidade de uso de métodos de ensino que permitam tornar o processo de ensino-aprendizagem mais efetivo. (SANTOS et al., 2008, p.3).

Em relação à estratégia de ensino, o currículo na área de ES (ACM/IEEE, 2004, p. 7 e 51, tradução nossa) ressalta a necessidade de mover-se para além do formato de aula expositiva. Neste sentido, é importante considerar a variação de técnicas de ensino e aprendizado. As abordagens mais comuns para ensinar ES incluem aulas expositivas, aulas de laboratório, entre outros.

Entretanto, o ensino da Engenharia de Software não requer apenas estudar a teoria usando livros, mais também proporcionar aos estudantes a experiência de questões mais técnica e de maneira mais efetiva, como por exemplo: a substituição de aulas expositivas por discussão de casos práticos (GNATZ et al., 2003, p.1, tradução nossa), dinâmicas de grupo, o uso de jogos (WANGENHEIM, 2012, p.22).

Prikladnicki et al. (2009, p.3-7) cita algumas estratégias participativas para o ensino de Engenharia de Software usadas em universidades para os cursos de Ciências da Computação e Sistemas de Informação aplicadas ao ensino da engenharia de software.

- *O uso de dinâmicas de grupo, educação à distância e atividades práticas:*

Voltado à disciplina de Gerência de Projeto de Software, tem como objetivo de aprendizagem apresentar os conceitos básicos de gerência de projetos de software, mesclando teoria e prática.

Tem como estratégia de ensino dinâmicas que exploram assuntos específicos com características chaves: diversificação nas técnicas de dinâmicas de grupo: uso de dinâmicas sobre gerencia de comunicação envolvendo raciocínio, mini fábrica de aviões e jogo de memória para trabalhar os conceitos do PMBOK. A dinâmica com aviões são utilizadas para introduzir conceitos de processo de gerência de projetos, e aulas práticas em laboratório, porque apesar de gerência de projetos ter um conteúdo bastante teórico, a disciplina é organizada de forma que os alunos explorem conceitos de forma prática, envolvendo a definição de escopo de projeto. Esta atividade obteve experiências positivas.

- *O uso de capstone projects e atividades práticas:*

Objetiva reforçar a compreensão dos conceitos, processos e técnicas da Engenharia de Software, estimular o aluno a ter a competência para aplicá-los com assistência na prática, visa à evolução de habilidades, como: comunicação, trabalho em equipe e resolução de problemas.

Sua estratégia de ensino é a internalização de conhecimento e habilidades pelo "aprender fazendo". Durante a disciplina é simulado um projeto de software, que inclui as fases de planejamento, execução (cobrindo as fases de análise de requisitos, design, implementação e testes) e a monitoração final.

O projeto se inicia na fase de planejamento, onde os alunos elaboram um plano de projeto, e em seguida iniciam as atividades técnicas seguindo o plano de projeto e o modelo de processo de desenvolvimento pré-definido baseado no modelo de ciclo de vida cascata. Na execução do projeto, as fases de análise de requisitos, design, implementação e os testes são realizados pelos alunos. Por fim, todos os dados referentes a prazo e esforços gastos nas atividades são avaliados e comparados com as estimativas no plano de projeto inicial, e apresentado os resultados parciais do projeto no final de cada fase.

O principal ponto forte observado pelos alunos após a estratégia ter sido aplicada, é o fato que a disciplina oferece a oportunidade de acompanhar a execução de um projeto de software do início ao fim com o objetivo de reunir todos os conhecimentos e habilidades aprendidos anteriormente durante o curso.

3.2. Jogos educacionais para engenharia de software

Jogos estão cada vez mais presentes como uma prática habitual no ensino e treinamento, sendo criado como uma atividade lúdica que é bastante motivadora no processo de ensino-aprendizado.

Desta forma, o uso de jogos para praticar, aprender e realizar atividades reais em ambientes realísticos pode melhorar o desempenho dos alunos, pois possibilita a vivência de experiências de aprendizagem que não são fornecidas de forma teórica. (MONSALVE et al., 2010, p.17)

A Engenharia de Software é uma área onde aspectos teóricos e aspectos práticos interagem, tornando fundamentais as decisões e experiências da prática no desenvolvimento de softwares de qualidade, econômicos, úteis e no prazo esperado. (MONSALVE et al., 2010, p.17).

Apesar disso, na área de engenharia de software esse método é pouco estudado. Um curso típico de engenharia de software consiste de aulas em que conceitos teóricos são passados aos alunos e exercitados por atividades em pequenos exemplos práticos. Mas não são satisfatoriamente cobertos nas atividades práticas.

Para contemplar esses aspectos práticos surgiram alguns jogos de tabuleiros, de cartas e simuladores para apoiar o ensino de engenharia de software, como:

SimulES: é um jogo de tabuleiro para ensino de Engenharia de Software, foi proposto com esse objetivo de exercitar, através de simulação, algumas situações e decisões do processo de desenvolvimento de software. O objetivo do jogo é gerir um projeto da melhor forma possível escolhendo as tarefas para os funcionários conforme grau de experiência e fadiga. O jogo começa com um orçamento e um tempo estimado para o projeto. Este jogo tem como finalidade estabelecer uma ligação entre a grande quantidade de conhecimento conceitual dado aos estudantes nas aulas e a pequena quantidade desse conhecimento aplicada em casos práticos (MONSALVE et al., 2010, p.19-20)

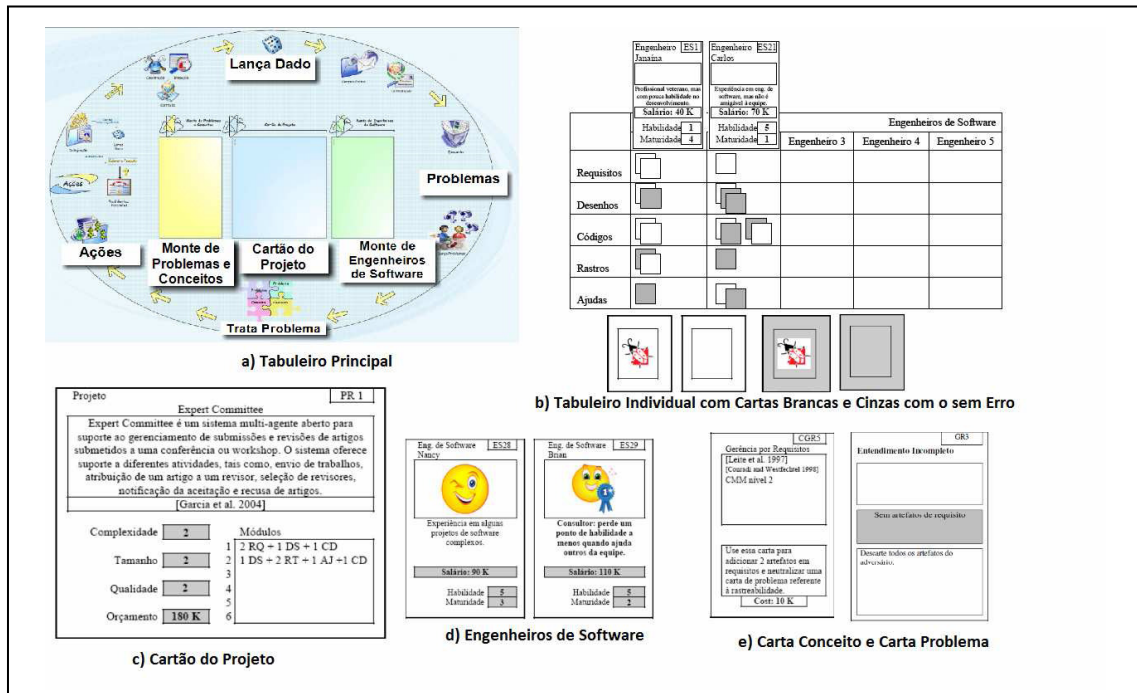


Figura 1 - Jogo SimSE.
Fonte: Monsalve et al., (2010)

SimSE: É um dos jogos mais conhecidos na área de Engenharia de Software, o SimSE é um simulador de processos de Engenharia de Software onde o papel do jogador é ser gestor de um projeto. O objetivo do jogo é gerir um projeto da melhor forma possível escolhendo as tarefas para os funcionários conforme grau de experiência e fadiga. O jogo começa com um orçamento e um tempo estimado para o projeto. Este jogo tem como finalidade estabelecer uma ligação entre a grande quantidade de conhecimento conceptual dado aos estudantes nas aulas e a pequena quantidade desse conhecimento aplicada em casos práticos (SimSE, NAVARRO E HOEK, 2009).

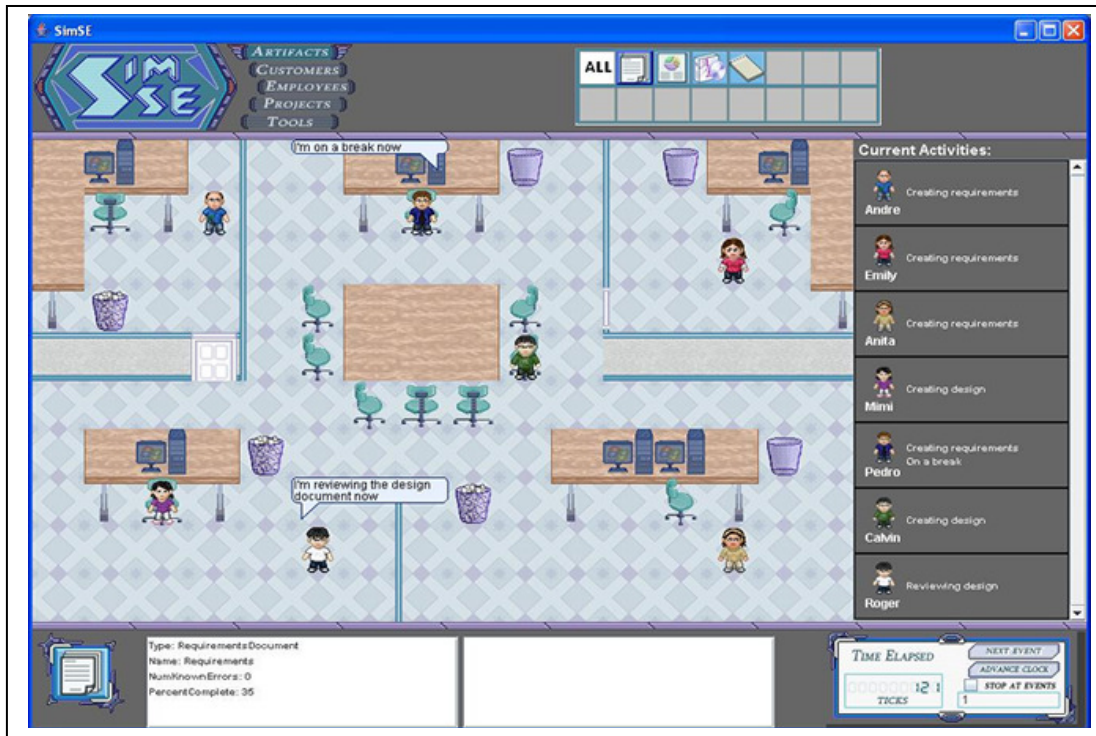


Figura 2 - Jogo SimSE.

Fonte: Navarro e Hoek (2009)

MO-SEProcess: é um jogo multijogador 3D de processo de Engenharia de Software baseado no jogo SimSE desenvolvido para o Second Life. O jogo tem como objetivo ensinar os princípios do processo de engenharia de software, simulando o desenvolvimento de um projeto de software de tamanho moderado (MO-SEProcess 2013).



Figura 3 - Jogo MO-SEProcess - Ambiente de jogo.

Fonte: MO-SEProcess 2013

PM Master: é um jogo de tabuleiro usado ao final de uma série de aulas apresentando conhecimento sobre o ciclo de vida de gerencia de projetos com o objetivo de revisar e fixar esse conhecimento. Usando o tabuleiro, os alunos avançam o numero de campos jogados com o dado e então respondem uma pergunta múltiplo-escolha sobre a determinada área de conhecimento em que pararam (indicada pelas cores dos campos). Se responderem corretamente pergunta, recebem um pedaço triangular da cor correspondente. Vence o jogo, o jogador que primeiro completar o circulo com os pedaços triangulares de todas as cores. (WANGENHEIM, 2012, p. 30-31).



Figura 4 - Jogo de tabuleiro PM Master.

Fonte: Wangenheim (2012)

Ilha dos Requisitos: é um exemplo de jogo online, voltado à revisão de conhecimento sobre engenharia de requisitos. Neste jogo, o aluno é representado como aventureiro que cai numa ilha deserta e que tem que resolver vários desafios, como, por exemplo, o aluno precisa ordenar corretamente fases do processo de engenharia de requisitos. (THIRY et al., 2010, p. 5).



Figura 5 - Tela do jogo "A Ilha dos Requisitos".

Fonte: Gonçalves et al, (2010).

Lego SCRUM City: Nesse jogo os alunos em grupo recebem userstories requisitando a construção de objetos como casas e veículos com Lego. E para isso, executam estes projetos aplicando todo o processo de SCRUM. (WANGENHEIM, 2012, p. 37).



Figura 6 - Jogo Lego SCRUM City

Fonte: Alexey Krivitsky (2011)

O jogo AprendES: é um jogo de cartas que tem como base o Simules, onde todos os conceitos de Engenharia de Software compreendidos pelo jogo são pré- determinados em cartões de projetos, o que não permite ao aluno vivenciar todas as etapas do processo de montagem do tabuleiro e apreender os fundamentos teóricos da disciplina. O jogo AprendES também permite a colaboração entre os membros da equipe, que são na verdade Engenheiros de Software, com o objetivo de finalizar o projeto. (FEITOSA; CAMPOS, 2010, p.3).

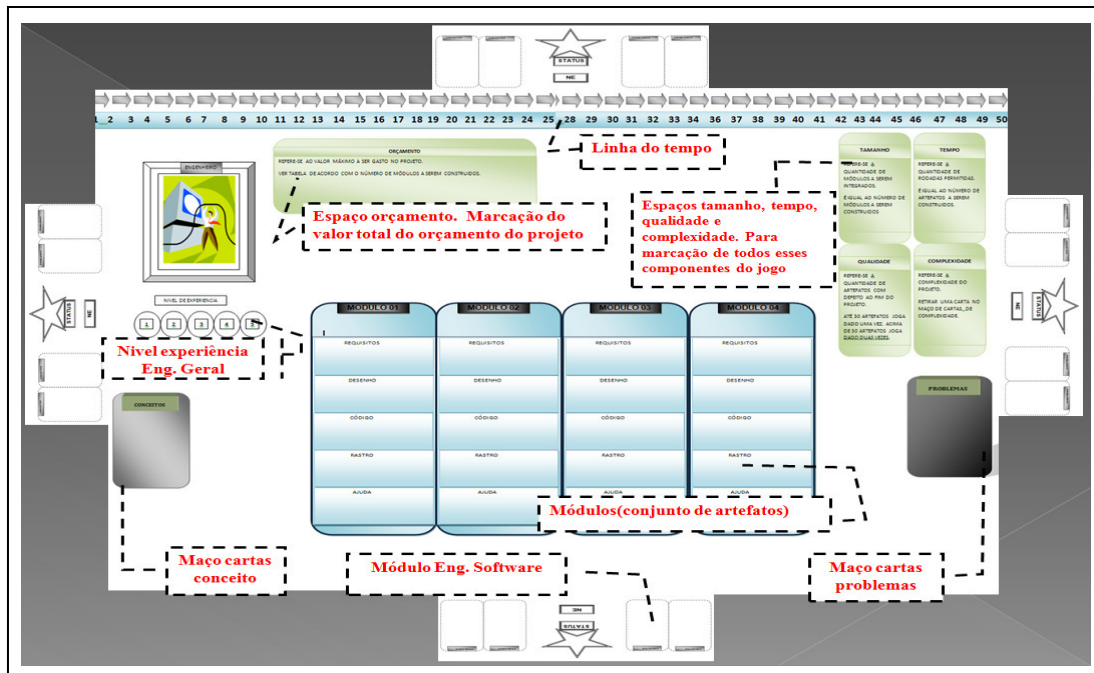


Figura 7 - Tabuleiro do Jogo AprendES

Fonte: Feitosa e Campos (2010)

4. ENGENHARIA DE SOFTWARE, UMA VISÃO GERAL

O software está cada vez mais presente na vida das pessoas. E há uma crescente demanda por softwares de qualidade (PRESSMAN, 2011, p.38).

Pressman (1995, p. 48) comenta:

O software tornou-se o elemento chave da evolução dos sistemas e produtos baseados em computador. No decorrer das últimas quatro décadas, o software evoluiu de uma ferramenta de análise de informações e de resolução de problemas especializada para uma indústria em si mesma. Mas logo a cultura e a história da “programação” criaram um conjunto de problemas que persistem até hoje. O software tornou-se um fator limitante na evolução dos sistemas baseados em computador.

A engenharia de software surgiu desta necessidade de aumento da qualidade dos produtos de software, bem como da diminuição de custos e riscos envolvidos em seu processo de desenvolvimento e da criação de rotinas que possam ser reaproveitadas em diversas fases deste processo, ou até em diferentes produtos. Pode-se dizer que a engenharia de software surgiu objetivando a utilização de princípios de engenharia sobre os já existentes processos de desenvolvimento de software (ENGHOLM JÚNIOR, 2010, p. 34-35).

A engenharia de software tem aplicação sobre todos os aspectos e estágios da criação de software, partindo das etapas iniciais de especificação até o momento em que o software está em uso e há a necessidade de realizar manutenção no mesmo, não se preocupando apenas com processos técnicos, e sim, vai além, viabilizando ou facilitando atividades de apoio à produção de software, como, por exemplo, a gerência de projetos (SOMMERVILLE, 2011, p. 5).

Pressman (1995, p 31) considera a engenharia de software como um rebento da engenharia de sistemas e de hardware, abrangendo um conjunto de três elementos essenciais: métodos, ferramentas e procedimentos.

Métodos proporcionam os detalhes de “como fazer” para construir o software, envolvendo um conjunto de tarefas incluindo, “planejamento e estimativa de projeto, análise de requisitos de software e de sistemas, projeto da estrutura de dados, arquitetura de programa de algoritmo de processamento, codificação, teste e manutenção.” (PRESSMAM, 1995, p.31).

As ferramentas fornecem apoio automatizado ou semi-automatizado aos métodos. (PRESSMAM, 1995, p32)

Os procedimentos da engenharia de software estabelece o elo e conserva juntos os métodos e ferramentas e possibilita o desenvolvimento coerente e oportuno do software. (PRESSMAM, 1995, p32)

Para Sommerville (2011, p.5) a abordagem sistemática usada na engenharia de software, às vezes, é chamada processo de software. “Um processo de software é um sequência de atividades que leva à produção de um produto de software”.

Existem quatro atividades fundamentais comuns a todos os processos de software (SOMMERVILLE, 2011, p.5-6). São elas:

1. Especificação de software, clientes e engenheiros definem o software a ser produzido.
2. Desenvolvimento de software, o software é projetado e programado.
3. Validação de software, verificação do software para garantir as exigências do cliente.
4. Evolução de software, modificação do software de acordo com os requisitos do cliente e do mercado.

5. VISÃO GERAL DO JOGO – PROPOSTA DE APRENSIZAGEM

SoftGame é um jogo de tabuleiro educacional direcionado para o ensino de engenharia de software.

Ele é composto por três níveis de complexidade: fácil, intermediário e difícil.

O jogo deve ter no mínimo dois jogadores ou duas equipes de jogadores.

O Jogo contém um tabuleiro, uma ampulheta, cento e três cartas, um dado e quatro peões.

5.1. Objetivo do jogo

Fazer o peão da própria equipe ser o primeiro a percorrer todo o trajeto do tabuleiro.

Os peões são movimentados quando os jogadores conseguem responder corretamente as questões das cartas da equipe adversária.

5.2 .Componentes Do Jogo

- O Tabuleiro

O tabuleiro é composto por cinquenta e quatro casas, sendo que treze casas são especiais (cor roxa), chamada de casa Hacker, trinta e nove são casas da cor verde. Sendo uma casa de "Início" e a ultima casa "Você Ganhou".

O tabuleiro tem a forma espiral, baseado no Modelo Espiral que é um modelo evolucionário de processo de software, é uma forma de realizar iterações mais organizadas, iniciando com protótipos pequenos e avançando para projetos maiores. (Wazlawick, 2013, p.36).

O jogo contém o desenho de uma ampulheta e um dado, que é somente ilustrativo e serve para posicionar os respectivos objetos para iniciar a partida.

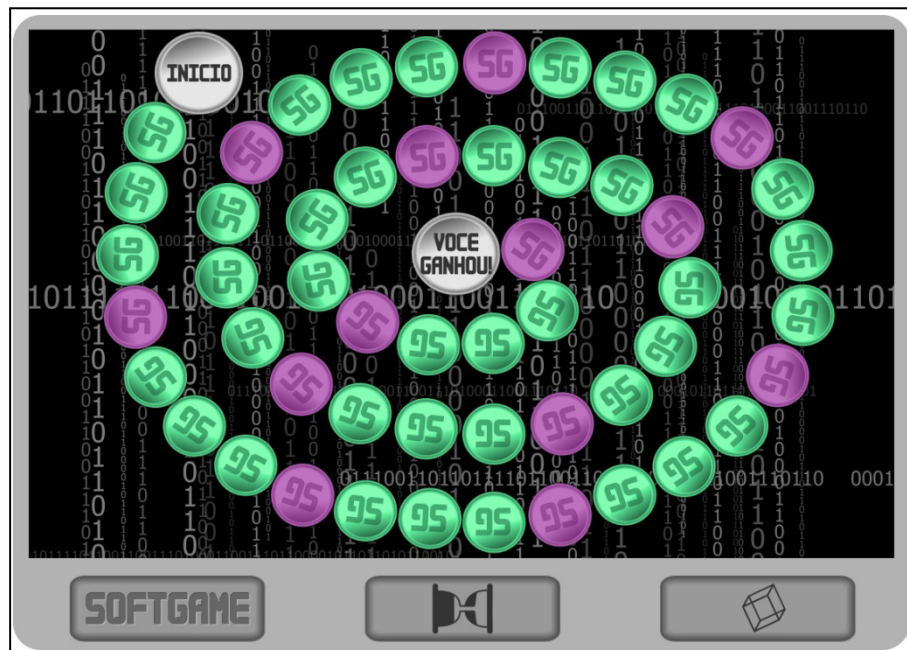


Figura 8 – Tabuleiro do jogo SoftGame.

Fonte: Autor (2016)

- A Ampulheta

No jogo a ampulheta serve como o contador do tempo da resposta de cada pergunta realizada, sendo que cada jogador tem no máximo 50 segundos para responder a pergunta.

- O Dado

Cada lado do dado tem um objetivo diferente:

Dois Lados possuem a letra **F**, ou seja, o jogador deverá responder a pergunta correspondente ao nível Fácil.

Um Lado possui a letra **I**, ou seja, o jogador deverá responder a pergunta correspondente ao nível Intermediário.

Um lado possui a letra **D**, ou seja, o jogador deverá responder a pergunta correspondente ao nível Difícil.

Um lado possui a frase “**Adversário Escolhe**”, neste caso o adversário escolhe qual nível de pergunta que o jogador da vez irá responder pela equipe.

Um lado possui a frase “**Perde/Ganha**”, o jogador da vez, escolhe o nível de pergunta que deseja responder, caso a resposta esteja correta, pontua e avança as casas, caso a resposta esteja errada, o adversário pontua e avança as casas no tabuleiro, sendo que o jogador da vez, não pontua e nem perde.

- As Cartas

O jogo contém 78 (setenta e oito cartas) com o símbolo Verde, correspondente à cor das casas do tabuleiro, em cada carta existem três níveis de perguntas e cada nível tem uma pontuação. A resposta correta à pergunta está sublinhada e em negrito na carta. No início de cada pergunta contém o nível destacado nas cores: **Verde = Fácil, Amarelo = Intermediário e Vermelho = Difícil.**

Níveis:

- **F** - Fácil: perguntas de Verdadeiro ou Falso – Vale 2 pontos.
- **I** - Intermediário: perguntas contendo 4 opções de resposta a,b,c,d sendo que somente uma é verdadeira - Vale 4 pontos.
- **D** - Difícil: Perguntas objetivas, sem alternativas, com única resposta – Vale 6 pontos.

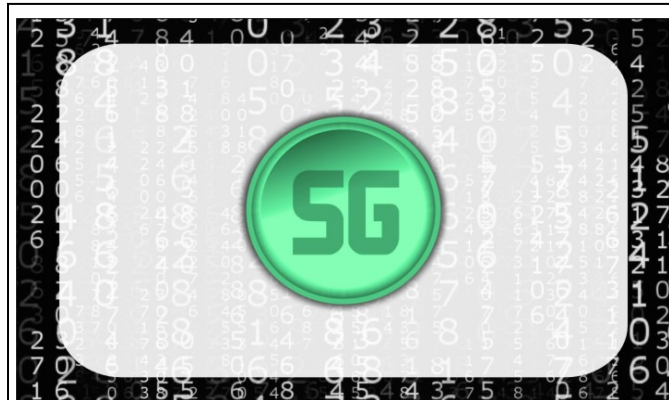


Figura 9 - Frente Carta.

Fonte: Autor (2016)

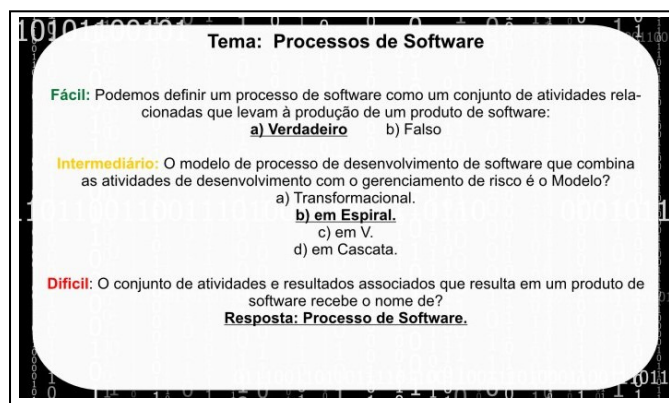


Figura 10 - Verso carta.

Fonte: Autor (2016)

- Carta Hacker

Esta carta abrange um conteúdo específico sobre os diagramas e conteúdos relacionados à UML, como também pode conter uma surpresa ao jogador. Ela será utilizada nas casas destacadas do tabuleiro, em cor Roxa, as casas Hacker. Totalizando 25 (vinte e cinco) cartas Hacker, com o símbolo na cor Roxa. Essa carta de perguntas, vale 3 (três) pontos, se respondida corretamente.

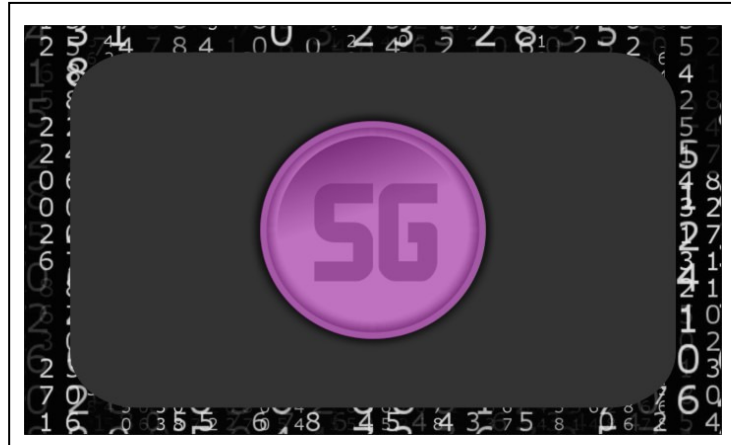


Figura 11 – Frente Carta Hacker.

Fonte: Autor (2016)

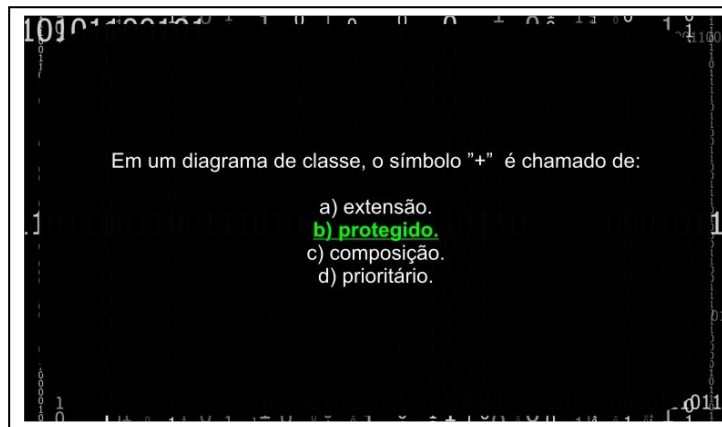


Figura 12 - Verso Carta Hacker.

Fonte: Autor (2016)

- Os Peões

Cada jogador terá direito a um peão, se a disputa for entre equipes, cada equipe terá apenas um peão. O jogo contém quatro peões, de diferentes cores.

6. DINÂMICA E PREPARAÇÃO DO JOGO

Para dar início ao jogo os jogadores devem ser divididos em duas equipes. Não há limite de jogadores para cada equipe. Se houver número ímpar de jogadores, não importa que uma equipe fique com um jogador a mais que a outra.

As cartas devem ser embaralhadas, e colocadas no centro da mesa.

Cada equipe escolhe um peão e o coloca na casa do tabuleiro marcada “INICIO”. Em seguida, segundo um critério qualquer, as equipes devem estabelecer qual o primeiro participante a jogar e qual a ordem de rodízio dos jogadores, para que a cada rodada, haja um participante diferente respondendo, assim todos participarão.

6.1 O Jogo

O jogador que iniciará a partida lançará o dado no tabuleiro e responderá a pergunta, que conforme o resultado deve pertencer ao nível em que o lado do dado caiu, o adversário deve executar a pergunta ao jogador.

Após a pergunta, a ampulheta é virada e o jogador tem 50 segundos para responder à equipe adversária.

Se o jogador (ou equipe) responder corretamente a pergunta, o peão se move em direção a casa "Você Ganhou", avançando o número de casas compatível com o número de pontos referente ao nível da pergunta.

Ele continua a jogar, lançando o dado, avançando o peão o número de casas respectivas à pontuação do nível da pergunta, fazendo um rodízio de jogadores e respondendo a uma nova carta. A carta anterior é colocada atrás das outras.

Vários peões podem ocupar uma mesma casa ao mesmo tempo. A cada jogada, é necessária a troca de jogadores da equipe, deve haver um rodízio na posição.

Se a pergunta não for respondida, ou respondida incorretamente o dado e a jogada passam para a próxima equipe.

A equipe que jogará, começa sua vez lançando o dado e respondendo a nova carta que o adversário comprar.

O jogo prossegue desse modo até que um dos peões complete todo o percurso.

6.1.1. O Juiz

Caso haja incompatibilidades ou duvidas quanto às respostas de nível Difícil, a decisão será do Juiz. Deve-se eleger um juiz, sendo na maioria das vezes, o professor, ou jogador escolhido por ambas as equipes. Este não participará da disputa e deve ser imparcial aos resultados.

6.1.2. O Vencedor

Para ganhar o jogo, uma equipe precisa levar seu peão até a última casa “Você Ganhou”.

6.2. Conteúdo do Jogo

Para a formulação das perguntas, foi dividido a Engenharia de software em cinco partes.

1. Engenharia de Software - Introdução

Envolve os conceitos básicos sobre o que é engenharia de software.

2. Engenharia de Requisitos

Engloba requisitos funcionais e não funcionais, especificação de requisitos, elicitação e análise de requisitos, validação de requisitos e gerenciamento de requisitos.

3. Produção de Software e modelos de Processo

Compreendem os conceitos e modelos de processos de softwares, os modelos de processo prescritivos e modelos ágeis, e também o processo unificado.

4. Planejamento e Gerencia de Projetos

Neste tópico compreende o planejamento, gerenciamento de Riscos, gerenciamento de projeto de software e gerenciamento de Configuração e Mudança.

5. Qualidade

Abrange sobre qualidade de software, testes de software, manutenção e evolução de software.

7. RESULTADOS OBTIDOS

Como a avaliação deve ser obtida a partir da percepção dos alunos, foi desenvolvido um questionário para a coleta de dados, o jogo foi aplicado a alunos da disciplina de Engenharia de Software II da Universidade Federal de Santa Catarina, foi avaliado um total de quatorze alunos, divididos em duas equipes. A partida obteve uma duração de aproximadamente cinquenta minutos, e após o término do jogo foi aplicado o questionário para avaliação.

O questionário foi inspirado no quadro 1 que é apresentado a seguir, ele possui itens sobre variados aspectos de avaliação, e são apresentados na forma de afirmações.

O formato de resposta para os itens selecionados no questionário do jogo SoftGame é baseado na escala Likert de 5 pontos, variando de -2 (discordo fortemente) até 2 (concordo fortemente).



Figura 13 – Alunos jogando SoftGame

Fonte: Autor (2016)

<p>MOTIVAÇÃO</p> <p>Atenção</p> <ul style="list-style-type: none"> - Houve algo interessante no início do jogo que capturou minha atenção. - O design da interface do jogo é atraente. <p>Relevância</p> <ul style="list-style-type: none"> - Ficou claro para mim como o conteúdo do jogo está relacionado com coisas que eu já sabia. - Eu gostei tanto do jogo que gostaria de aprender mais sobre o assunto abordado por ele. - O conteúdo do jogo é relevante para meus interesses. - Eu poderia relacionar o conteúdo do jogo com coisas que já vi, fiz ou pensei. - O conteúdo do jogo será útil para mim. <p>Confiança</p> <ul style="list-style-type: none"> - O jogo foi mais difícil de entender do que eu gostaria. - O jogo tinha tanta informação que foi difícil identificar e lembrar dos pontos importantes - O conteúdo do jogo é tão abstrato que foi difícil manter a atenção nele. - As atividades do jogo foram muito difíceis. - Eu não consegui entender uma boa parcela do material do jogo. <p>Satisfação</p> <ul style="list-style-type: none"> - Completar os exercícios do jogo me deu um sentimento de realização. - Eu aprendi algumas coisas com o jogo que foram surpreendentes ou inesperadas. - Os textos de feedback depois dos exercícios, ou outros comentário do jogo, me ajudaram a sentir recompensado pelo meu esforço. - Eu me senti bem ao completar o jogo. <p>EXPERIÊNCIA DO USUÁRIO</p> <p>Imersão</p> <ul style="list-style-type: none"> - Eu não percebi o tempo passar enquanto jogava. - Eu perdi a consciência do que estava ao meu redor enquanto jogava. - Me senti mais no ambiente do jogo do que no mundo real. - Me esforcei para ter bons resultados no jogo. 	<ul style="list-style-type: none"> - Houve momentos em que eu queria desistir do jogo. - Me senti estimulado a aprender com o jogo <p>Desafio</p> <ul style="list-style-type: none"> - Eu gostei do jogo e não me senti ansioso ou entediado. - O jogo me manteve motivado a continuar utilizando-o. -Minhas habilidades melhoraram gradualmente com a superação dos desafios - O jogo oferece novos desafios num ritmo apropriado. - Este jogo é adequadamente desafiador para mim, as tarefas não são muito fáceis nem muito difíceis. <p>Habilidade / Competência</p> <ul style="list-style-type: none"> - Me senti bem sucedido. - Eu alcancei rapidamente os objetivos do jogo. - Me senti competente. - Senti que estava tendo progresso durante o desenrolar do jogo. <p>Interação Social</p> <ul style="list-style-type: none"> - Senti que estava colaborando com outros colegas. - A colaboração no jogo ajuda a aprendizagem. - O jogo suporta a interação social entre os jogadores. <p>Divertimento</p> <ul style="list-style-type: none"> - Eu gosto de utilizar este jogo por bastante tempo. - Quando interrompido, fiquei desapontado que o jogo tinha acabado. - Eu jogaria este jogo novamente. - Algumas coisas do jogo me irritaram. - Fiquei torcendo para o jogo acabar logo. - Achei o jogo meio parado. <p>CONHECIMENTO</p> <ul style="list-style-type: none"> - Depois do jogo consigo lembrar de mais informações relacionadas ao tema apresentado no jogo. - Depois do jogo consigo compreender melhor os temas apresentados no jogo. - Depois do jogo sinto que consigo aplicar melhor os temas relacionados com o jogo.
---	---

Quadro 1 – Questões afirmativas para avaliação de jogos educacionais

Fonte: Wangenheim (2010)

Este modelo de avaliação de jogos educacionais foi criado com o intuito de fornecer uma base para tornar possível a avaliação de jogos educacionais de forma mais sistemática e padronizada, além de facilitar a própria avaliação. (WANGENHEIM, 2012, p.88).

Algumas perguntas foram retiradas do quadro 1 para a confecção do questionário aplicado (Anexo 1), sendo que para a avaliação dos resultados as perguntas foram divididas em três níveis, inspirado no modelo de avaliação Kirkpatrick (WANGENHEIM, 2012, p.85), sendo:

1) MOTIVAÇÃO

Motivação para aprender é um elemento essencial em qualquer sistema educacional (KELLER, 2009 apud SAVI et al., 2010, p.3), por isso a aprendizagem precisa realizada com cuidado para despertar um nível adequado de motivação nos estudantes, sendo em sala de aula ou em um ambiente virtual.

Um dos métodos existentes que objetiva empregar estratégias motivacionais no projeto de materiais instrucionais é o modelo ARCS de John Keller.

ARCS é um acrônimo que identifica quatro categorias de estratégias importantes para que se consiga motivar os alunos na aprendizagem: Atenção, Relevância, Confiança e Satisfação (Attention, Relevance, Confidence, Satisfaction) (KELLER, 2009 apud SAVI et al., 2010, p.3).

Para estas quatro categorias foi selecionado perguntas para realizar a avaliação, sendo elas:

Atenção

- ✓ A variação (conteúdo ou atividade) ajudou a me manter atento ao jogo.
- ✓ O design do jogo é atraente (cartas e tabuleiros).

Relevância

- ✓ O conteúdo do jogo é relevante para os meus interesses
- ✓ O funcionamento deste jogo está adequado ao meu jeito de aprender.

Confiança

- ✓ Foi fácil entender o jogo e começar a utilizá-lo como material de estudo.

Satisfação

- ✓ Estou satisfeito porque sei que terei oportunidades de utilizar na prática coisas que aprendi com o jogo
- ✓ No decorrer do jogo senti confiança de que estava aprendendo.

2) EXPERIÊNCIA DO USUÁRIO

A experiência do usuário contempla a interação do indivíduo com o produto por inteiro, considerando também pensamentos, sentimentos, prazer e demais percepções que resultam da interação. (SAVI et al., 2010, p.5).

Existe uma expectativa de que a utilização de jogos produza experiências positivas (GÁMEZ, 2009 apud SAVI et al., 2010, p.3). Os jogadores se envolvem e se aprofundam no ambiente do jogo, interagem com outros objetos, e isso vem acompanhado de emoções, que são consideradas parte fundamental do uso de jogos.

Existem alguns conceitos consideravelmente importantes para avaliar a experiência do usuário, e para eles também foi selecionado algumas perguntas para realizar a avaliação (SAVI et al., 2010, p.5), são eles:

Imersão

- ✓ Eu não percebi o tempo passar enquanto jogava, quando vi o jogo acabou.
- ✓ Temporariamente esqueci as minhas preocupações do dia-a-dia, fiquei totalmente concentrado no jogo.

Interação

- ✓ O jogo promove momentos de cooperação e/ou competição entre as pessoas que participam.
- ✓ Me diverti junto com outras pessoas durante o jogo.

Diversão

- ✓ Me diverti com o jogo.
- ✓ Eu recomendaria este jogo para meus colegas.
- ✓ Gostaria de utilizar este jogo novamente.

Desafio

- ✓ Este jogo é adequadamente desafiador para mim, as tarefas não são muito fáceis nem muito difíceis.

Competência

- ✓ Tive sentimentos positivos de eficiência no desenrolar do jogo.

3) APRENDIZAGEM

A aprendizagem é medida pela realização das duas perguntas abaixo, buscando compreender se o jogo é eficaz para a aprendizagem do aluno em comparação com outras atividades da disciplina e também medir se o jogo contribuiu de maneira positiva ou não na aprendizagem da disciplina. Segue abaixo as questões aplicadas:

- ✓ O jogo foi eficiente para minha aprendizagem, em comparação com outras atividades da disciplina.
- ✓ O jogo contribuiu para minha aprendizagem na disciplina.

Depois da coleta dos dados pelo questionário, foi especificados valores para fazer referencia a escala Likert e para a confecção dos gráficos e melhor entendimento dos resultados.

A escala Likert de 5 pontos varia de -2 até 2, sendo -2(discordo fortemente), -1 (discordo), 0 (neutro), 1(concordo) e 2 (concordo fortemente), para esta avaliação a escala assume os valores de 1 (discordo fortemente), 2 (discordo), 0 (neutro), 3 (concordo) e 4 (concordo fortemente).

Cada pergunta foi avaliada individualmente (gráficos em anexo) e em seguida agrupadas nos seus respectivos níveis para uma segunda avaliação, essa segunda avaliação é a média ponderada dos resultados das perguntas avaliadas individualmente, que será apresentada a seguir.

7.1. Análise do questionário aplicado

A análise a seguir oferece como resultado uma pontuação sobre diferentes aspectos a respeito da avaliação do jogo educacional SoftGame.

O jogo Softgame foi avaliado, e recebeu um feedback muito positivo dos alunos. Em termos de motivação, se observado na figura abaixo, a grande maioria dos alunos, concorda fortemente com todos os itens, incluindo satisfação, confiança de que estão aprendendo, relevância do conteúdo e a atenção.

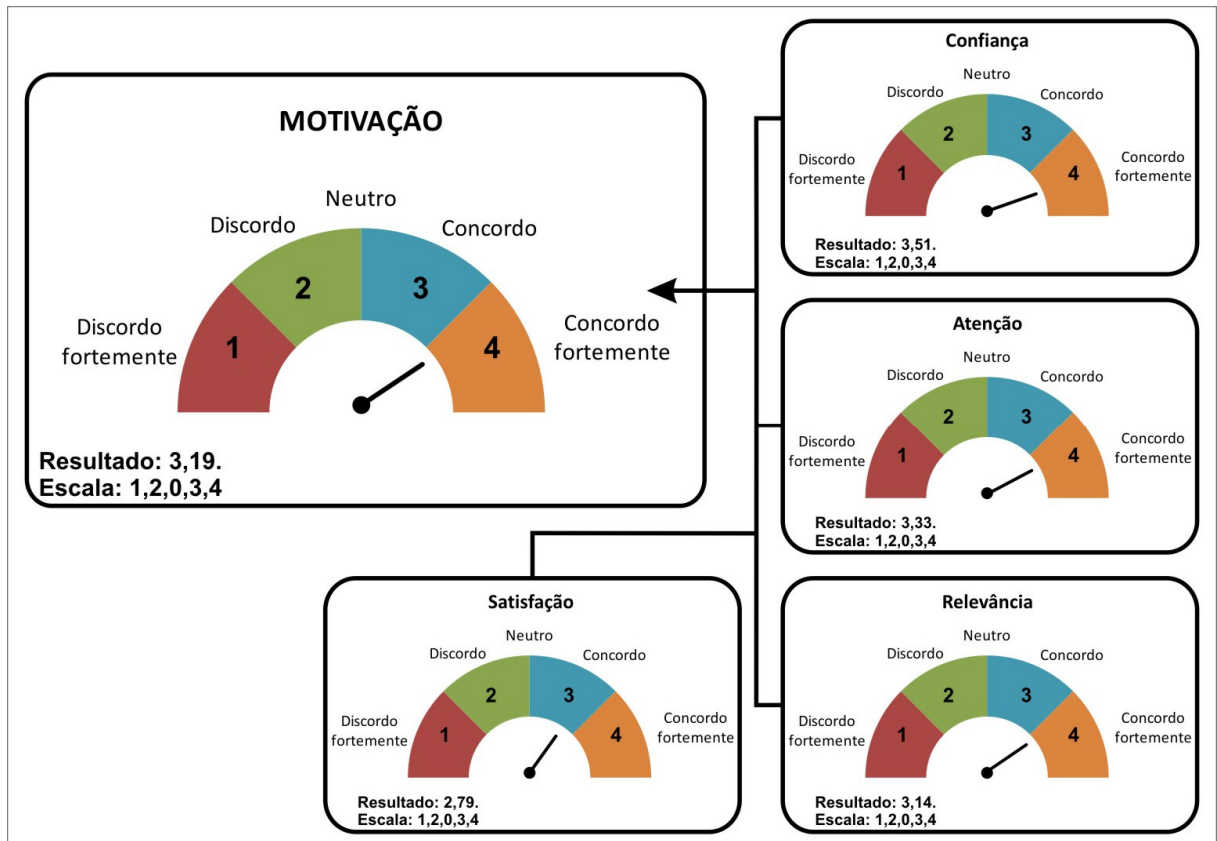


Figura 14 - Avaliação Nível Motivação

Fonte: Autor (2016).

Em termos de experiência de usuário, os alunos avaliaram o jogo mais positivo ainda, levando em consideração aspectos como competência, diversão, desafio, interação social e imersão. Eles especialmente reconhecem que se divertem com outros alunos estimulados pela interação social evocada pelo jogo.

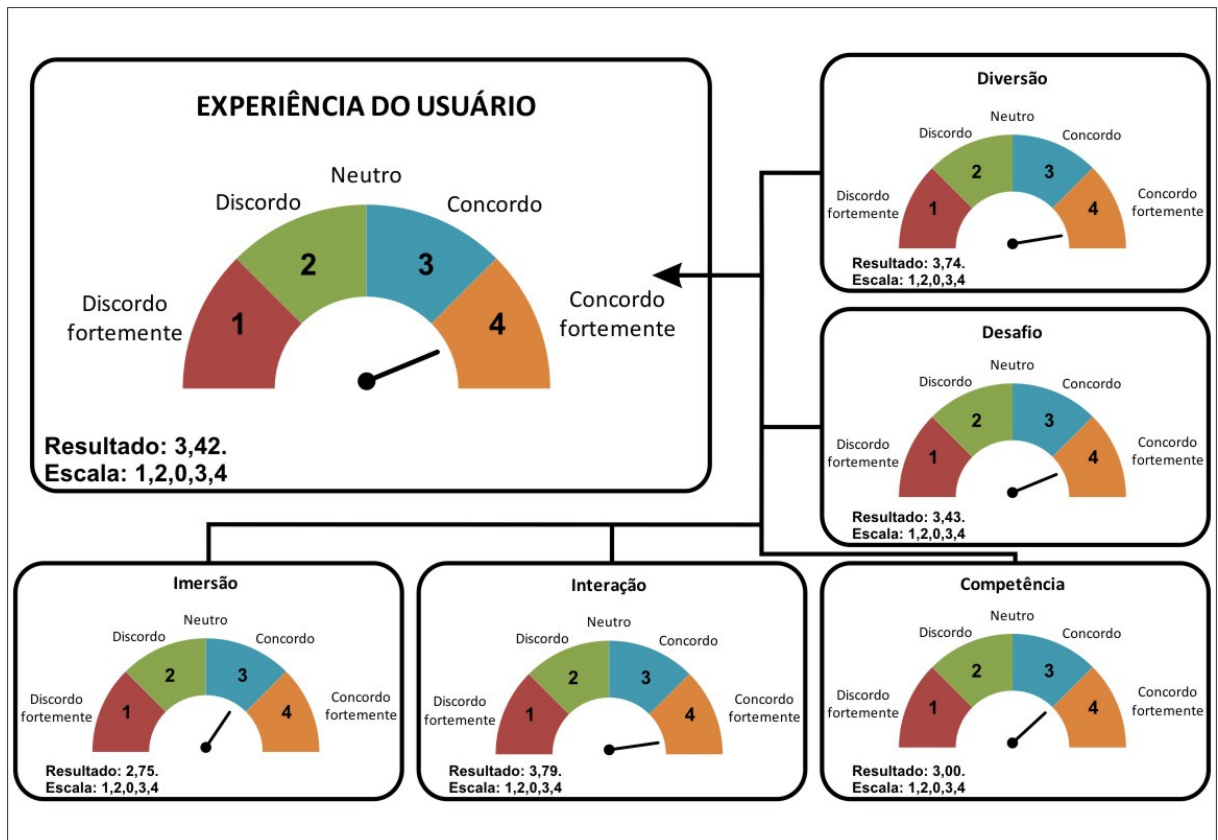


Figura 15 - Avaliação Nível Experiência do Usuário

Fonte: Autor (2016).

O jogo SoftGame também atribuiu na aprendizagem, como indicam as respostas dos alunos, que numa auto-avaliação reconhecem que o jogo contribuiu na aprendizagem do conteúdo da disciplina, e também se comparado com outras atividades.

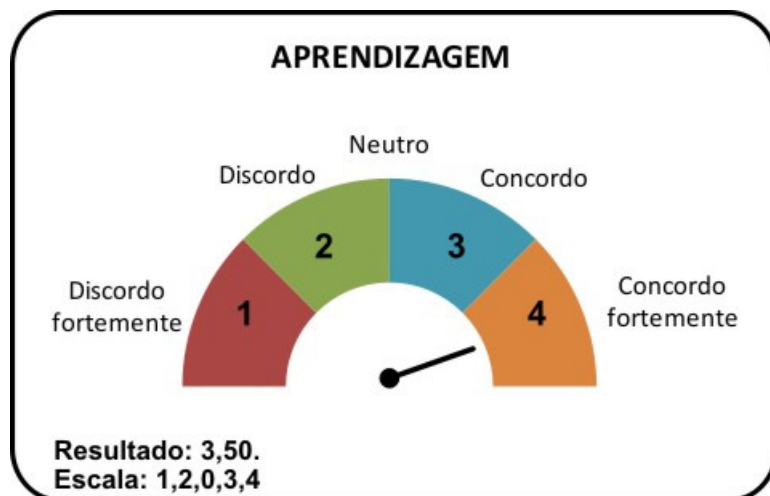


Figura 16 - Avaliação Nível Aprendizagem

Fonte: Autor (2016)

8. CONSIDERAÇÕES FINAIS

O aprendizado de Engenharia de Software é desafiante tanto para alunos quanto para professores. O ensino da E.S enfrenta o desafio de levar conceitos teóricos vistos em sala de aula para experiências práticas. Para suprir tal deficiência, podem ser utilizados jogos de apoio ao ensino, com foco em conceitos específicos.

Este artigo relata a experiência de uso do jogo SoftGame em sala de aula na Universidade Federal de Santa Catarina, e esta experiência contribuiu para o aprendizado dos alunos, na medida em que eles combinam a tarefa de aprender com o entretenimento proporcionado por um jogo, além de ser uma alternativa viável para atingir esta meta. Este equilíbrio é capaz de motivar os alunos e melhorar o nível de interesse pela disciplina.

O jogo desenvolvido, além de focar no conhecimento através de cartas com perguntas e respostas sobre as Engenharia de Software, reforça em estratégias através das cartas Hacker, dados, ampulheta e peões, tornando o jogo mais dinâmico para tentar despertar o interesse do aluno em jogar e consequentemente aprender com o jogo.

Após as avaliações, podemos considerar ainda mais os jogos como um grande aliado para o aprendizado, agregando conhecimento no processo de aprendizagem, tornado as aulas divertidas e proporcionando ao aluno o aprendizado através da “brincadeira”.

REFERÊNCIAS

ACM/IEEE.Computer Science Curriculum, 2008. Disponível: <https://www.acm.org/education/curricula/ComputerScience2008.pdf>. Acessado em: 08 de maio de 2016.

ACM/IEEE.Software Engineering Curriculum.Guidelines for Undergraduate Degree Programs in Software Engineering, 2004. Disponível: [sites.computer.org/ccse/SE2004Volume.pdf](https://www.acm.org/education/curricula/SoftwareEngineering2004.pdf). Acessado em: 08 de maio de 2016.

AILLOIS, Roger. **Os jogos e os homens: a máscara e a vertigem**. Lisboa: Cotovia, 1990
AMBRÓSIO, M.J.;Pablo SCHOEFFEL, P.; **Jogo De Tabuleiro Para Apoio Ao Ensino De Gerenciamento De Projetos Baseado No Guia Pmbok**. UDESC – Universidade do estado de Santa Catarina, 2014.

BOMTEMPO, Edda. **A brincadeira de faz-de-conta: lugar do simbolismo, da representação, do imaginário**. In: KISHIMOTO, Tizuko. **Jogo, brinquedo, brincadeira e a educação**. 7. ed, São Paulo: Cortez, 2003.

ENGHOLM, Hélio Jr. **Engenharia de Software na Prática**. São Paulo. Novatec, 2010.

FEITOSA, A.C; CAMPOS, G.M. M. **AprendES: uma proposta lúdica para auxiliar o ensino da Engenharia de Software**. Universidade do Estado do Rio Grande do Norte (UERN), 2010.

GNATZ, M., KOF.L., PRILMEIER, F., SEIFERT, T.**A Practical Approach of Teaching Software Engineering**, Proc. 16th Conf. Software Eng. Education and Training, pp. 120–128, 2003. Disponível: <https://www.broy.in.tum.de/publ/papers/CSEET.pdf>. Acessado em:15 de maio de 2016.

GONÇALVES, R.Q.; THIRY, M; ZOUCAS, A. **Avaliação da Aprendizagem em Experimentos com Jogo Educativo de Engenharia de Requisitos**. Universidade do Vale do Itajaí (UNIVALI) – Campus São José. São José. 2010.

HUIZINGA, Johan. **Homo Ludens:o jogo como elemento da cultura**. 8ed. Perspectiva. São Paulo. 2014.

KISHIMOTO, T. M. **O jogo e a educação infantil**. Pioneira. São Paulo.1994

LEAL, T. F. **Jogos: alternativas didáticas para brincar alfabetizando (ou alfabetizar brincando?)**. In: Alfabetização: apropriação do sistema de escrita alfabética. Belo Horizonte: Autêntica, 2005.

MACEDO, L.; PETTY, A.L.S.; PASSOS, N.C. 2000. **Aprender com jogos e situações problemas**. Porto Alegre: Artes Médicas Sul.

MACHADO, L.V., **MARIINHA: Jogo Educacional Como Objeto De Aprendizagem Para Crianças Do Ensino Fundamental Da Escola Municipal Mariinha Rocha**. Balsas/Ma. Universidade de Balsas – UNIBALSAS, Balsas, 2010.

MONSALVE, E.S.; WERNECK, V. M. B.; LEITE, J.C.S.P. **SimulES-W: Um Jogo para o Ensino de Engenharia de Software**. Pontifícia Universidade Católica de Rio de Janeiro (PUC-Rio), Rio de Janeiro, Brasil Universidade do Estado do Rio de Janeiro (UERJ), Rio de Janeiro, Brasil, 2010.

MORAIS, A.G; ALBUQUERQUE, E.B.C; LEAL, T.F. **Alfabetização apropriação do sistema de escrita alfabética**. 2005. Universidade Federal de Pernambuco, Belo Horizonte 2005.

MORATORI, P.B. **Por Que Utilizar Jogos Educativos No Processo De Ensino Aprendizagem?** Universidade Federal do Rio De Janeiro. Rio de Janeiro. 2003.

MOURA, Manoel O. **A séria busca no jogo: do lúdico na matemática**. In: KISHIMOTO, Tizuko. Jogo, brinquedo, brincadeira e a educação. 7ª ed. São Paulo: Cortez, 2003.

PRESSMAN, Roger S. **Engenharia de Software**. Pearson. Makson Books, São Paulo,1995.

PRESSMAN, Roger S. **Engenharia de Software**. Pearson. 7. ed. AMGH, Porto Alegre, 2011.

PRIKLADNICKI, R; ALBUQUERQUE, A.B.; WANGENHEIM, C. G. ;Reinaldo CABRAL, R. **Ensino de Engenharia de Software: Desafios, Estratégias de Ensino e Lições Aprendidas**. UNIVALI, 2009.

SALEN, Katie; ZIMMERMAN, Eric. **Regras do jogo: fundamentos do design de jogos**. Edgard Blucher São Paulo.2012.

SANTOS, Daniela Silva dos. **O Papel dos Jogos de Tabuleiro na Aprendizagem**. Disponível em <http://meuartigo.brasilecola.com/pedagogia/o-papel-dos-jogos-tabuleiro-naaprendizagem.htm>. Acessado em 07 de maio de 2016.

SANTOS, S.M.P.2006. **Educação, Arte e Jogo**. Rio de Janeiro: Vozes.

SAVI, R.; WANGENHEIM, V.G.C.; ULBRICHT, V.; VANZIN, T. **Proposta de um Modelo de Avaliação de Jogos Educacionais**. 2010. Artigo - CINTED-UFRGS, 2010.

SILVEIRA, R. S; BARONE, D. A. C. **Jogos Educativos computadorizados utilizando a abordagem de algoritmos genéticos**. Universidade Federal do Rio Grande do Sul. Instituto de Informática. Curso de Pós-Graduação em Ciências da Computação. 1998.

SOARES, Michele dos Santos. **Projetos de Jogos Educativos 2D de Aventura usando Lua**. 2012. 63pg. Dissertação– Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2012.

SOMMERVILLE, Ian. **Engenharia de Software**. 9ed. Pearson Prentice Hall. São Paulo. 2011.

TAROUCO, L. M. R.; FABRE, Marie-christine Julie Mascarenhas ; ROLAND, Letícia Coelho ; KONRATH, Mary Lúcia Pedroso . **Jogos educacionais**. RENOTE. Revista Novas Tecnologias na Educação, Porto Alegre, v. 2, n. 1, p. 1-7, 2004.

WANGENHEIM, C.G V.; WANGENHEIM, A.V. 2012. **Ensinando computação com jogos**. Florianópolis: Bookess.

WANGENHEIM, C.G.; KOCHANISKI, D., SAVI, R. **Revisão Sistemática sobre Avaliação de Jogos Voltados para Aprendizagem de Engenharia de Software no Brasil**. Programa de Pós-Graduação em Ciência da Computação/ Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC) Florianópolis – SC.2009.

WAZLAWICK, R.S, **Engenharia de software: conceitos e práticas**. Rio de Janeiro, Elsevier, 2013.

ZUCARELLI, I; COUTO. L. **Jogo de tabuleiro em incentivo à alimentação infantil**. Universidade São Judas Tadeu, São Paulo, 2013.

ANEXOS

ANEXO 1 – Questionário de Avaliação Aplicado

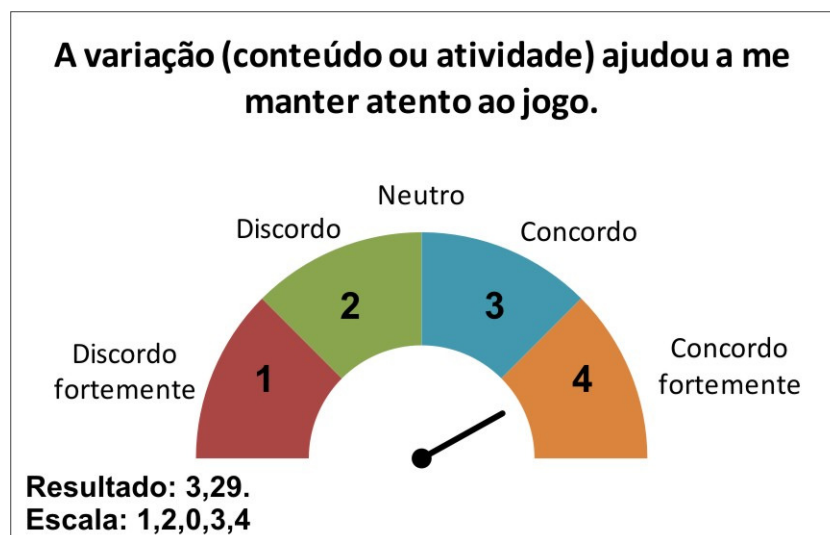
QUESTIONÁRIO JOGO SOFTGAME

Por favor, circule um numero de acordo com o quando você concorda ou discorda de cada afirmação abaixo.

Afirmações	Sua avaliação			Comentários sobre a questão
A variação (conteúdo ou atividade) ajudou a me manter atento ao jogo.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Me diverti com o jogo.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Este jogo é adequadamente desafiador para mim, as tarefas não são muito fáceis nem muito difíceis.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Me diverti junto com outras pessoas durante o jogo	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Eu não percebi o tempo passar enquanto jogava, quando vi o jogo acabou.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Temporariamente esqueci as minhas preocupações do dia-a-dia, fiquei totalmente concentrado no jogo.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
O jogo foi eficiente para minha aprendizagem, em comparação com outras atividades da disciplina.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
O jogo contribuiu para minha aprendizagem na disciplina.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
O design do jogo é atraente (cartas e tabuleiros).	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
O conteúdo do jogo é relevante para os meus interesses.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
O funcionamento deste jogo está adequado ao meu jeito de aprender.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Foi fácil entender o jogo e começar a utilizá-lo como material de estudo.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
No decorrer do jogo senti confiança de que estava aprendendo.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Estou satisfeito porque sei que terei oportunidades de utilizar na prática coisas que aprendi com o jogo.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
O jogo promove momentos de cooperação e/ou competição entre as pessoas que participam.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Tive sentimentos positivos de eficiência no decorrer do jogo.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Eu recomendaria este jogo para meus colegas.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Gostaria de utilizar este jogo novamente.	Discordo Fortemente	-2-1 0 +1 +2	Concordo Fortemente	
Qual a sua opinião sobre o jogo?				
Deixe sua sugestão de melhoria ou comentário.				

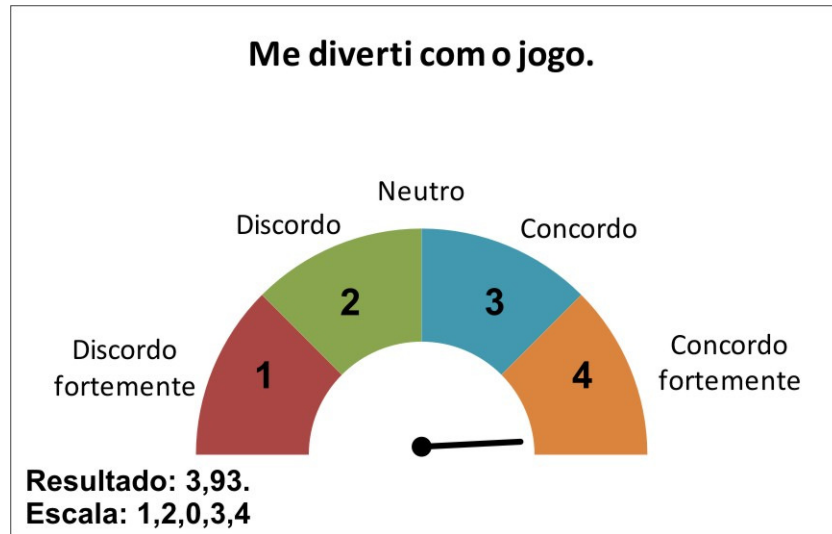
APÊNDICE

APÊNDICE 1 – Gráfico 1



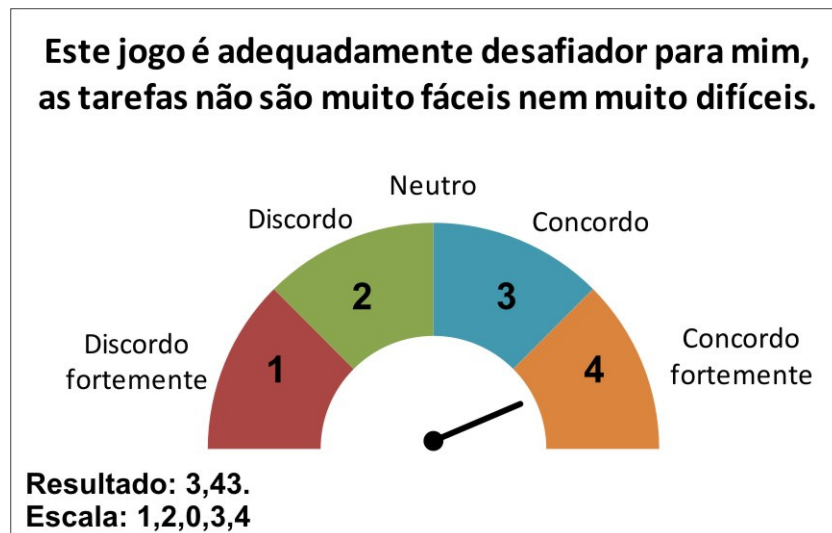
Fonte: Autor (2016)

APÊNDICE 2 – Gráfico 2



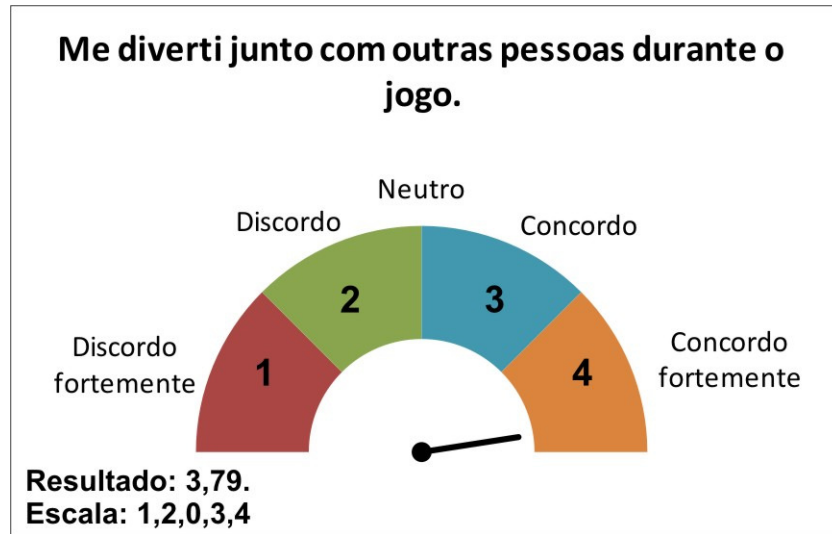
Fonte: Autor (2016)

APÊNDICE 3 – Gráfico 3



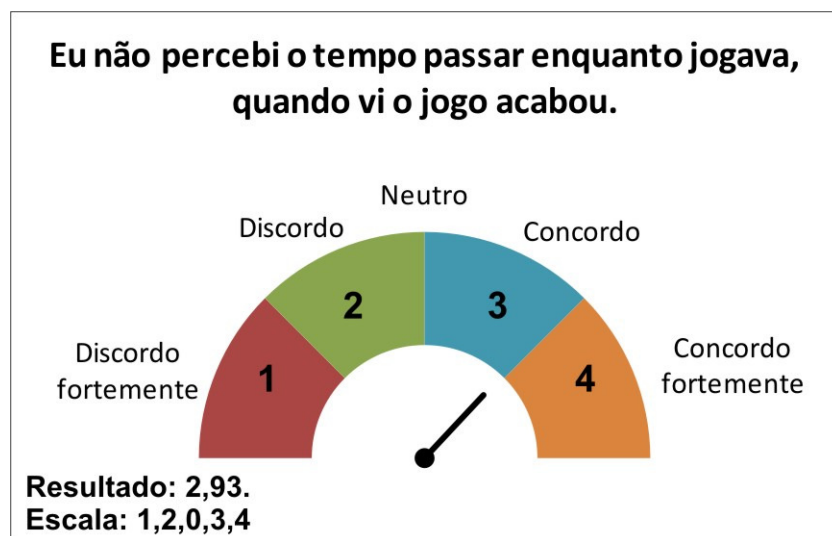
Fonte: Autor (2016)

APÊNDICE 4 – Gráfico 4



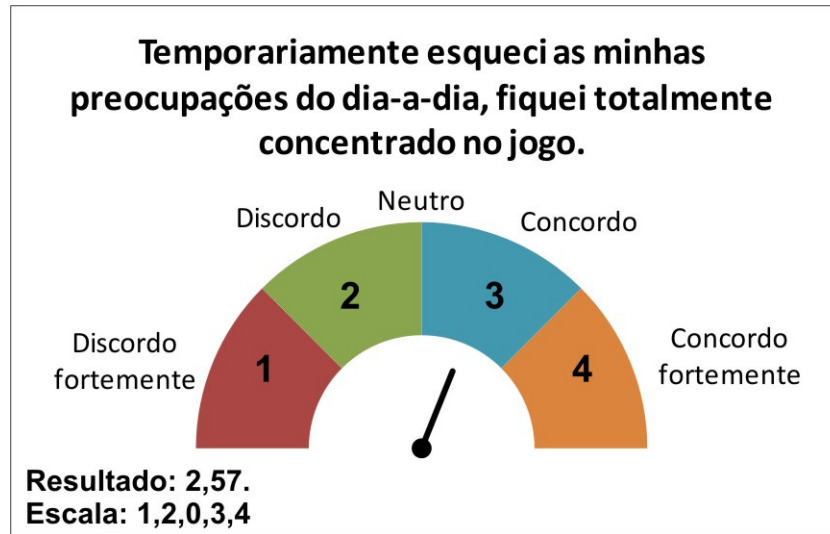
Fonte: Autor (2016)

APÊNDICE 5 – Gráfico 5



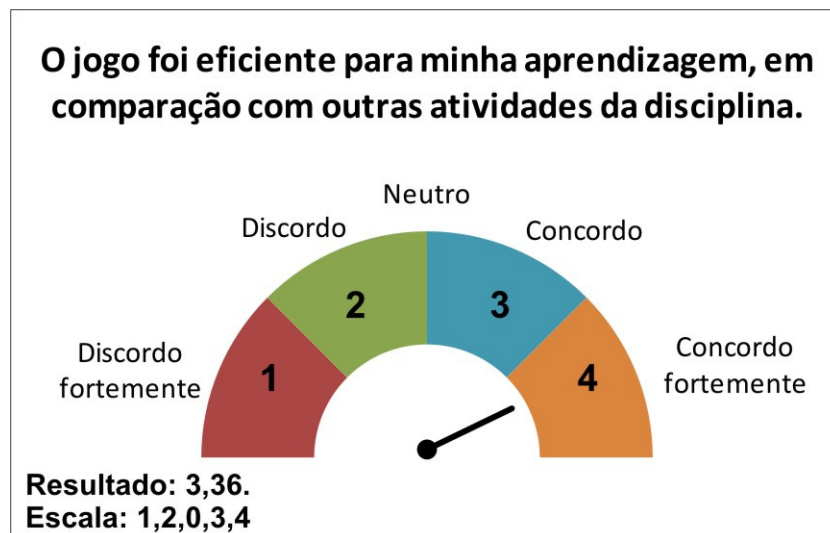
Fonte: Autor (2016)

APÊNDICE 6 – Gráfico 6



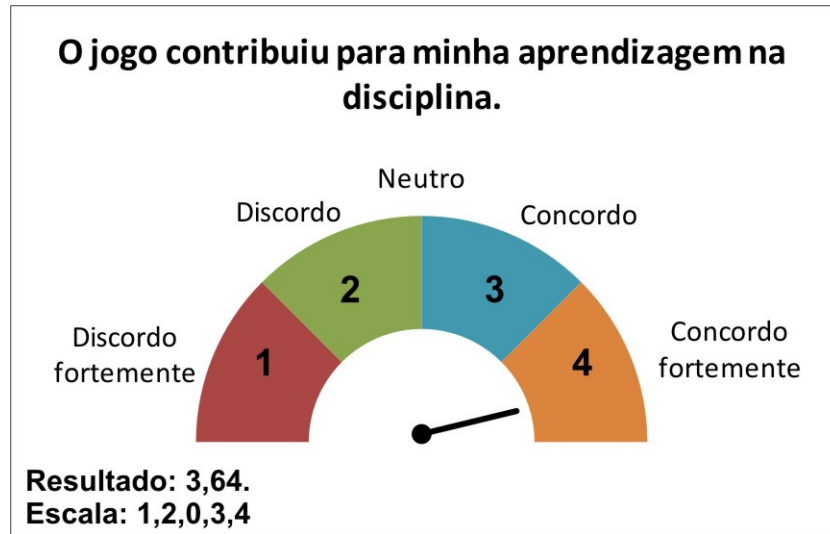
Fonte: Autor (2016)

APÊNDICE 7 – Gráfico 7



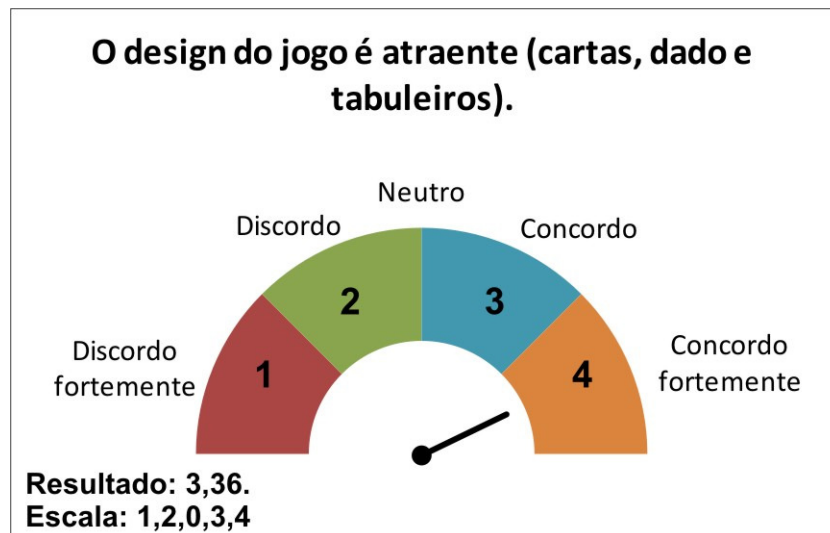
Fonte: Autor (2016)

APÊNDICE 8 – Gráfico 8



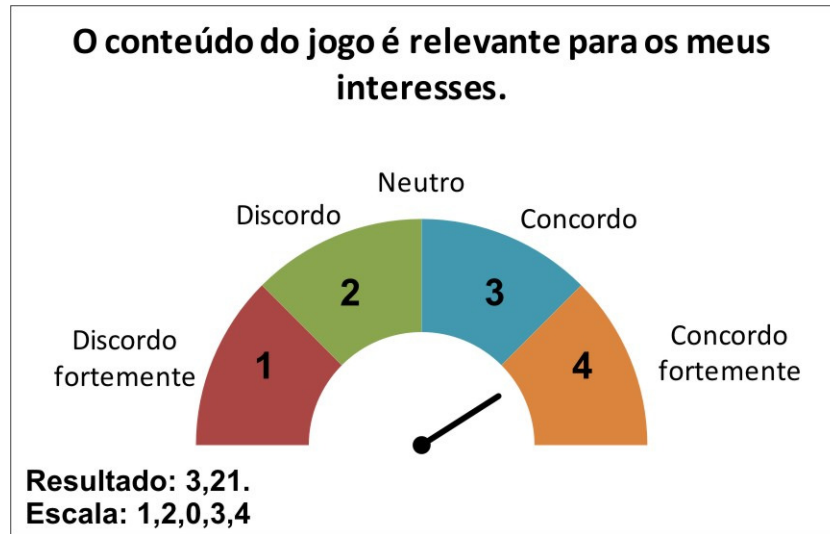
Fonte: Autor (2016)

APÊNDICE 9 – Gráfico 9



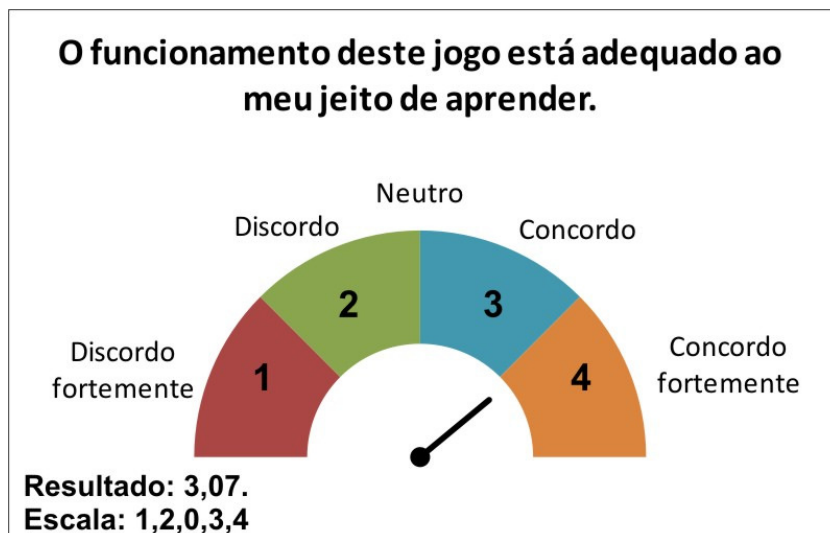
Fonte: Autor (2016)

APÊNDICE 10 – Gráfico 10



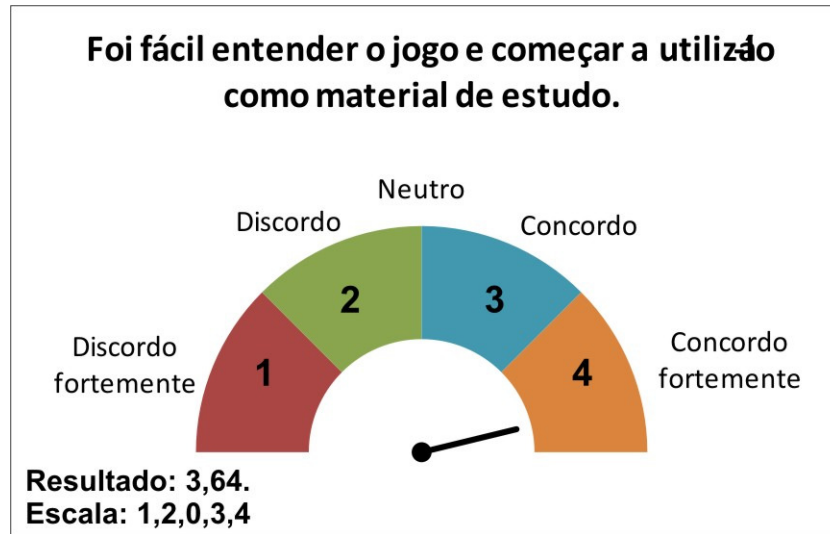
Fonte: Autor (2016)

APÊNDICE 11 – Gráfico 11



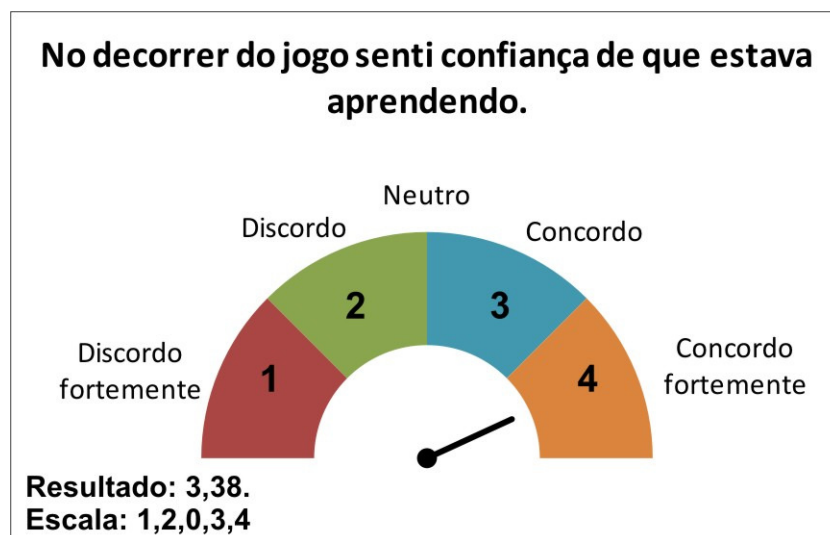
Fonte: Autor (2016)

APÊNDICE 12 – Gráfico 12



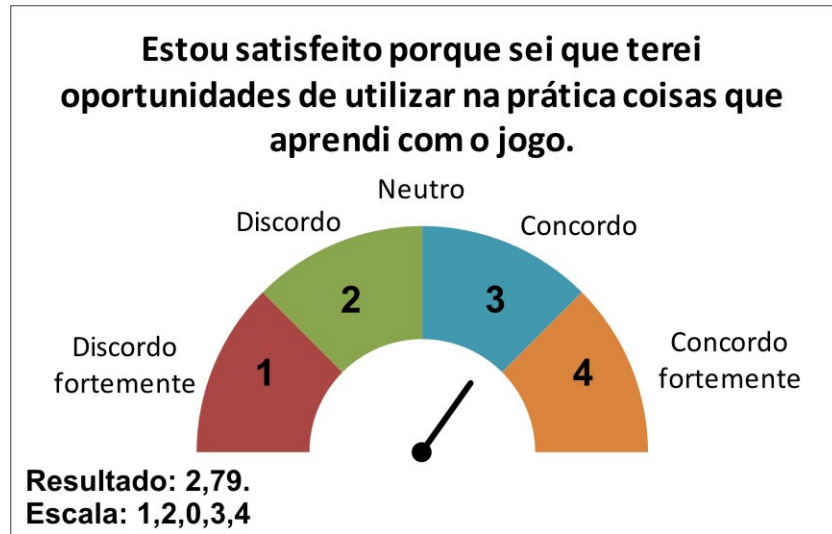
Fonte: Autor (2016)

APÊNDICE 13 – Gráfico 13



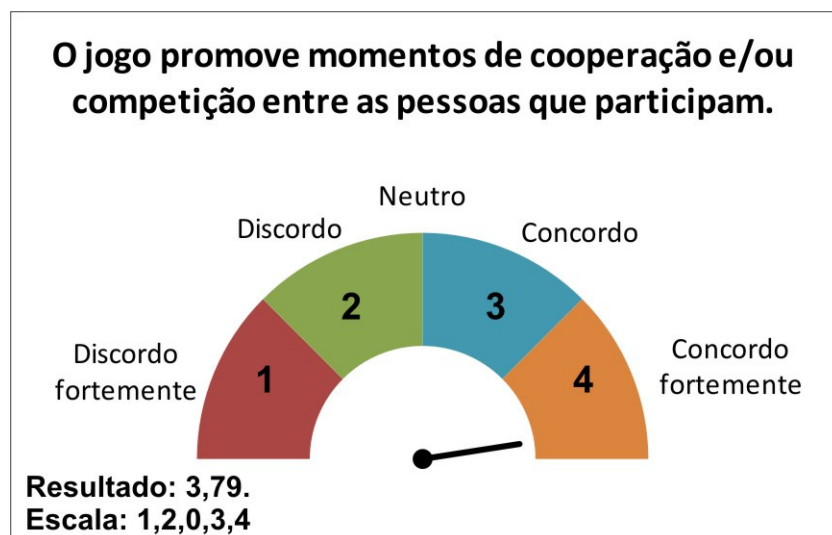
Fonte: Autor (2016)

APÊNDICE 14 – Gráfico 14



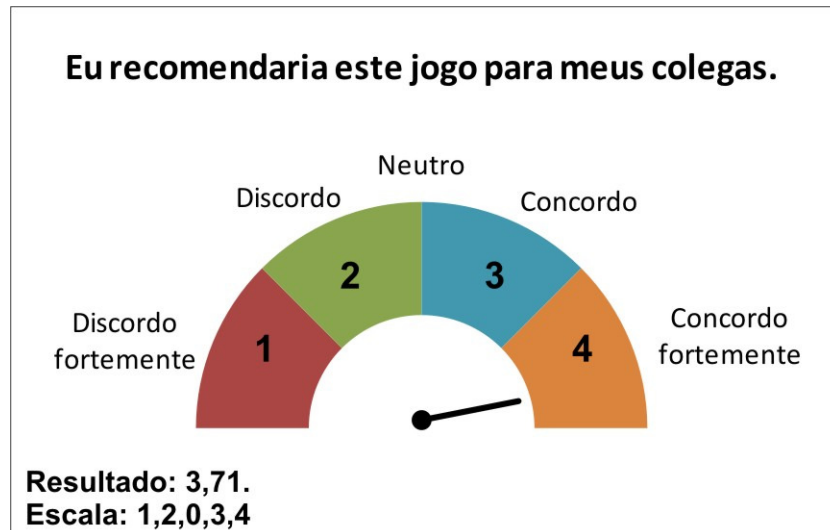
Fonte: Autor (2016)

APÊNDICE 15 – Gráfico 15



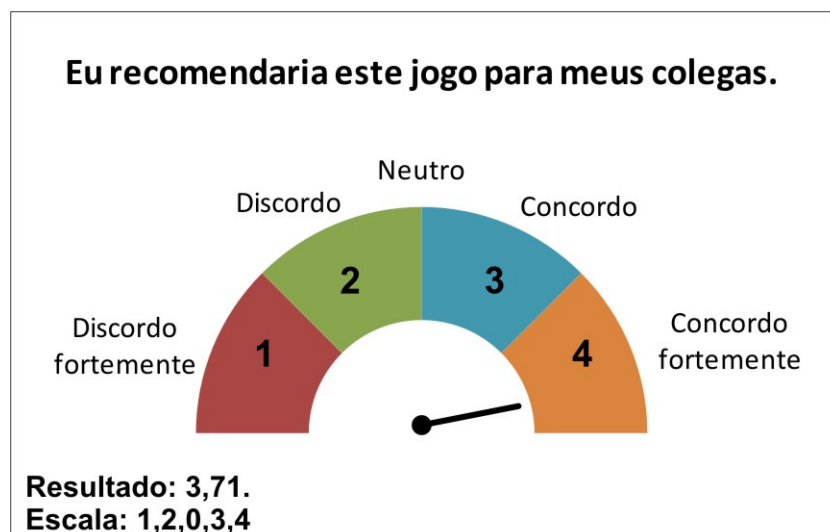
Fonte: Autor (2016)

APÊNDICE 16 – Gráfico 16



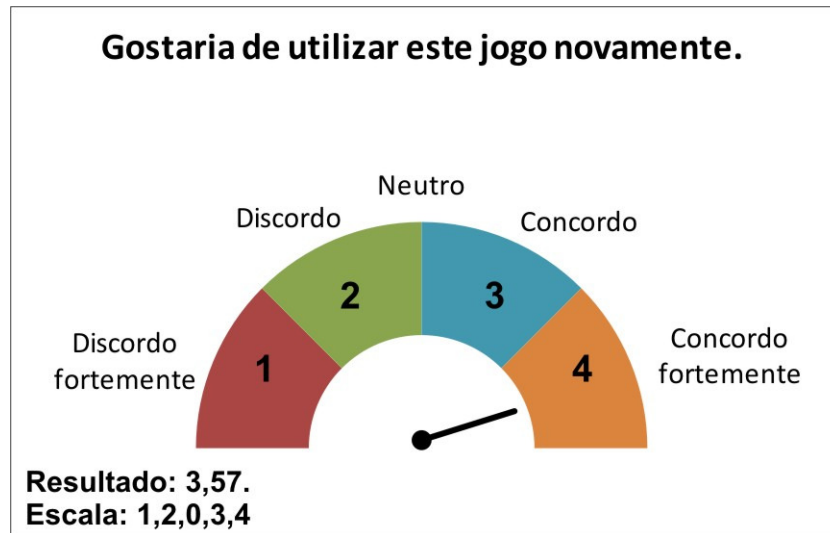
Fonte: Autor (2016)

APÊNDICE 17 – Gráfico 17



Fonte: Autor (2016)

APÊNDICE 18 – Gráfico 18



Fonte: Autor (2016)

APÊNDICE 19 – Alunos da Disciplina Engenharia de Software II - UFSC



Fonte: Autor (2016)

APÊNDICE 20 – Manual SoftGame

MANUAL DO JOGADOR



SoftGame é um jogo de tabuleiro educacional direcionado para o ensino de Engenharia de Software.

Regras Do Jogo:

Número de Participantes: para 2 ou mais jogadores, ou para 2 equipes.

Conteúdo:

– 1 tabuleiro; 1 ampulheta; 103 cartas; 1 dado; 4 peões.

Níveis:

- **F** - Fácil: perguntas de Verdadeiro ou Falso – **Vale 2 pontos.**
- **I** - Intermediário: perguntas contendo 4 opções de resposta a,b,c,d sendo que somente uma é verdadeira - **Vale 4 pontos.**
- **D** - Difícil: Perguntas objetivas, sem alternativas, com única resposta – **Vale 6 pontos.**

Componentes Do Jogo:

- **O Tabuleiro:** O tabuleiro é composto por cinquenta e quatro casas, sendo que treze casas são especiais (cor roxa), chamada de casa Hacker, trinta e nove são casas da cor verde. Sendo uma casa de "Início" e a última casa "Você Ganhou". O jogo contém o desenho de uma ampulheta e um dado, que é somente ilustrativo e serve para posicionar os respectivos objetos para iniciar a partida.

- **A Ampulheta:** No jogo a ampulheta serve como o contador do tempo da resposta de cada pergunta realizada, sendo que cada jogador tem no máximo 50 segundos para responder a pergunta.

- **O Dado:** Cada lado do dado tem um objetivo diferente:
Dois Lados possuem a letra **F**, ou seja, o jogador deverá responder a pergunta correspondente ao nível Fácil.
Um Lado possui a letra **I**, ou seja, o jogador deverá responder a pergunta correspondente ao nível Intermediário.
Um lado possui a letra **D**, ou seja, o jogador deverá responder a pergunta correspondente ao nível Difícil.
Um lado possui a frase "**Adversário Escolhe**", neste caso o adversário escolhe qual nível de pergunta que o jogador da vez irá responder pela equipe.

Um lado possui a frase "**Perde/Ganha**", o jogador da vez, escolhe o nível de pergunta que deseja responder, caso a resposta esteja correta, pontua e avança as casas, caso a resposta esteja errada, o adversário pontua e avança as casas no tabuleiro, sendo que o jogador da vez, não pontua e nem perde.

- **Cartas:** O jogo contém 78 (setenta e oito cartas) com o símbolo Verde, correspondente a cor das casas do tabuleiro, em cada carta existem três níveis de perguntas e cada nível tem uma pontuação. A resposta correta à pergunta está sublinhada e em negrito na carta. No início de cada pergunta contém o nível destacado nas cores:

Verde = Fácil, Amarelo = Intermediário e Vermelho = Difícil.

- **Cartas "Especiais" Hacker:** Esta carta abrange um conteúdo específico sobre os diagramas e conteúdos relacionados à UML, como também pode conter uma surpresa ao jogador. Ela será utilizada nas casas

APÊNDICE 21 – Manual SoftGame – Verso

destacadas do tabuleiro, em cor Roxa, as casas Hacker. Totalizando 25 (vinte e cinco) cartas Hacker, com o símbolo na cor Roxa. Essa carta de perguntas, vale 3 pontos, se respondida corretamente.

- **Os Peões:** Cada jogador terá direito a um peão, se a disputa for entre equipes, cada equipe terá apenas um peão. O jogo contém quatro peões, de diferentes cores.

Objetivo Do Jogo:

Fazer o peão da própria equipe ser o primeiro a percorrer todo o trajeto do tabuleiro.

Os peões são movimentados quando os jogadores conseguem responder corretamente as questões das cartas da equipe adversária.

Dinâmica e Preparação Do Jogo:

Para dar início ao jogo os jogadores devem ser divididos em duas equipes. Não há limite de jogadores para cada equipe. Se houver número ímpar de jogadores, não importa que uma equipe fique com um jogador a mais que a outra.

As cartas devem ser embaralhadas, e colocadas no centro da mesa.

Cada equipe escolhe um peão e o coloca na casa do tabuleiro marcada "INICIO". Em seguida, segundo um critério qualquer, as equipes devem estabelecer qual o primeiro participante a jogar e qual a ordem de rodízio dos jogadores, para que a cada rodada, haja um participante diferente respondendo, assim todos participarão.

O Jogo:

O jogador que iniciará a partida lançará o dado no tabuleiro e responderá a pergunta, que conforme o resultado deve pertencer ao nível em que o lado do dado caiu, o adversário deve executar a pergunta ao jogador.

Após a pergunta, a ampulheta é virada e o jogador tem 50 segundos para responder à equipe adversária.

Se o jogador (ou equipe) responder corretamente a pergunta, o peão se move em direção a casa "Você Ganhou", avançando o número de casas compatível com o número de pontos referente ao nível da pergunta.

Ele continua a jogar, lançando o dado, avançando o peão o número de casas respectivas à pontuação do nível da pergunta, fazendo um rodízio de jogadores e respondendo a uma nova carta. A carta anterior é colocada atrás das outras.

Vários peões podem ocupar uma mesma casa ao mesmo tempo. A cada jogada, é necessário a troca de jogadores da equipe, deve haver um rodízio na posição.

Se a pergunta não for respondida, ou respondida incorretamente o dado e a jogada passam para a próxima equipe.

A equipe que jogará, começa sua vez lançando o dado e respondendo a nova carta que o adversário comprar.

O jogo prossegue desse modo até que um dos peões complete todo o percurso.

O Juiz:

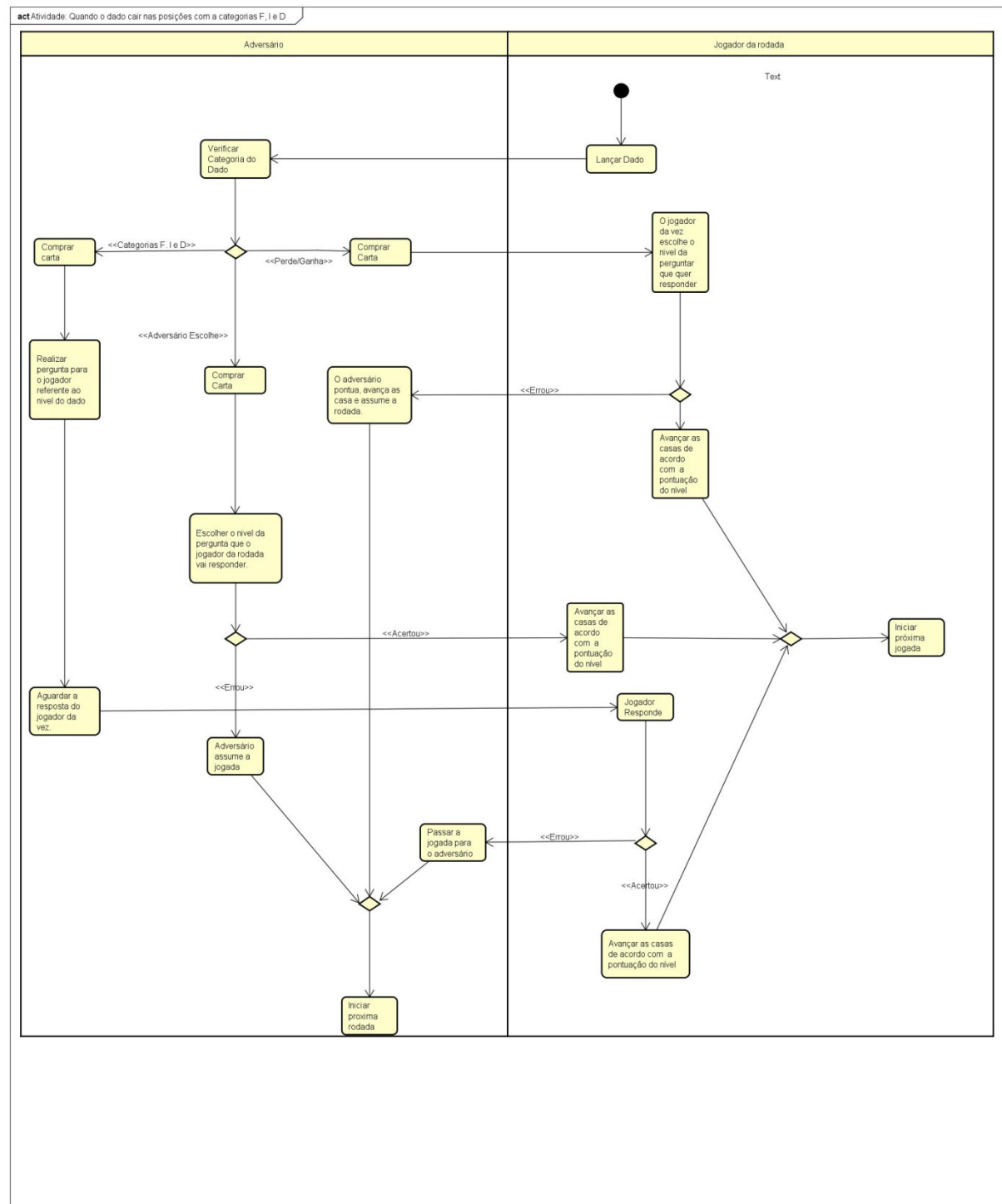
Caso haja incompatibilidades ou dúvidas quanto às respostas de nível Difícil, a decisão será do Juiz. Deve-se eleger um juiz, sendo na maioria das vezes, o professor, ou jogador escolhido por ambas as equipes. Este não participará da disputa e deve ser imparcial aos resultados.

O Vencedor:

Para ganhar o jogo, uma equipe precisa levar seu peão até a última casa "Você Ganhou".

Bom Divertimento e Bons Estudos!

APÊNDICE 22 – Diagrama de atividade do jogo SoftGame



Fonte: Autor (2016)

APÊNDICE 23 – Pergunta das Cartas - Tema: Fundamentos da Engenharia de software

CARTA 1

1 - Engenharia de software é uma disciplina de engenharia que se preocupa com todos os aspectos da produção do software.

a) Verdadeiro b) Falso

2 - Existem muitos tipos de software. Não existe um método ou uma técnica universal de engenharia de software que se aplique a todos. No entanto, há três aspectos gerais que afetam vários tipos de software:

- a) **Heterogeneidade, Mudança de negocio e social, Segurança e confiança.**
- b) Aceitabilidade, Eficiência, Confiança e proteção.
- c) Manutenibilidade, Ergonomia e Segurança.
- d) Ergonomia, Mudança social e Confiança.

3 - Quais são os cinco atributos importantes que todo software profissional deve possuir?

Resposta: Manutenibilidade, confiança, proteção, eficiência e aceitabilidade.

CARTA 2

1 - As ideias fundamentais da engenharia de software são universalmente aplicáveis para todos os tipos de desenvolvimento de sistemas. Esses fundamentos incluem processos de software, confiança, proteção, requisitos e reuso.

a) Verdadeiro b) Falso

2 - As ideias básicas da engenharia de software são aplicáveis a todos os tipos de sistema de software. Estes fundamentos incluem:

- a) **Processos de software gerenciados, confiança e proteção, engenharia de requisitos e reuso de software.**
- b) Especificação e desenvolvimento, manutenibilidade, validação e evolução do software.
- c) Processos de software, validação e evolução, e engenharia de requisitos.
- d) Processos de software gerenciados, especificação e desenvolvimento, eficiência e aceitabilidade.

3 – Qual o princípio que recomenda que um elemento que compõe o design de software deve ser representado apenas por suas características essenciais, permitindo assim, a distinção de outros elementos por parte do observador e, a consequente representação de um elemento do design mais simples, uma vez que detalhes desnecessários são descartados. **Resposta: Abstração.**

CARTA 3

1-Podemos afirmar que a Engenharia de Sistemas se preocupa com todos os aspectos do desenvolvimento de sistemas computacionais, incluindo engenharia de hardware, software e processo, enquanto a engenharia de software é uma parte específica desse processo mais genérico.

a) Verdadeiro b) Falso

2 - O software é um elemento de sistema lógico, e não físico. Portanto, o software tem características que são consideravelmente diferentes do hardware. Qual das opções abaixo não é considerada característica do software:

- a) O software é desenvolvido ou projetado por engenharia.
- b) O software não se “desgasta”;
- c) A maioria dos softwares é feita sob medida em vez de ser montada a partir de componentes existentes.
- d) Os custos do software não estão concentrados no trabalho de engenharia, podendo ser geridos como projetos manufaturados.**

3 - Qual a diferença mais importante entre o desenvolvimento de um produto genérico de software e o desenvolvimento de software sob demanda.

Resposta: Software genérico tem como característica ser desenvolvido de um forma padrão, sem um publico alvo ou algumas especificações de um cliente, ao contrario de um software sob demanda que é desenvolvido sob encomenda por um cliente particular.

CARTA 4

1 -Podemos afirmar que não existe um processo único e ideal para desenvolvimento de software, porque cada sistema tem suas particularidades. Porém, usualmente, podem-se agrupar os sistemas de acordo com certas características e então definir modelos de processos mais adequados a elas.

a) Verdadeiro b) Falso

2 -Do ponto de vista da engenharia de software, os sistemas podem ser classificados da seguinte forma:

- a) Genérico, Sob Encomenda, De tempo real, Científico, Jogos, Artificial.
- b) Básico, De tempo real, Comercial, Científico, Embutido, Pessoal, Jogos, Inteligência Artificial.**
- c) Genérico, Comercial, Embarcado, Científico, Jogos e Inteligência Artificial.
- d) Básico, Sob Encomenda, embarcado e Embutido, científico e de engenharia e Artificial.

3 - Referente aos mitos de Software, a frase "Após a edição do programa e a sua colocação em fun-

cionamento, o trabalho está terminado." É um mito do? **Resposta: Mito do Profissional**

CARTA 5

1 - A engenharia de software engloba processos, métodos e ferramentas. Um de seus focos é a produção de software de alta qualidade a custos adequados.

a) Verdadeiro b) Falso

2 - Considere: é uma disciplina que se ocupa de todos os aspectos da produção de software, desde os estágios iniciais de especificação do sistema até a manutenção desse sistema, depois que ele entrou em operação. Seu principal objetivo é fornecer uma estrutura metodológica para a construção de software com alta qualidade. A definição refere-se:

- a) ao ciclo de vida do software.
- b) à programação orientada a objetos.
- c) à análise de sistemas.
- d) à engenharia de requisitos.
- e) à engenharia de software.**

3 – A engenharia de software é baseada em camadas, com foco na qualidade. Essas camadas são?

Resposta: ferramentas, métodos e processo.

CARTA 6

1 - Entre os desafios enfrentados pela engenharia de software estão lidar com sistemas legados, atender à crescente diversidade e atender às exigências quanto a prazos de entrega reduzidos.

a) Verdadeiro b) Falso

2 - A Engenharia de Software é uma disciplina que se ocupa de todos os aspectos da produção de software, desde os estágios iniciais de especificação do sistema até a sua manutenção. A Engenharia de Software adota métodos que

a) são um conjunto de atividades, cuja meta é o desenvolvimento ou a evolução do software.

b) são uma representação simplificada de um processo de software, apresentada a partir de uma perspectiva específica.

c) são abordagens de estruturadas para o desenvolvimento de software, que incluem modelos de sistemas, notações, regras, recomendações de projetos e diretrizes de processos

d) se ocupam da teoria e dos fundamentos de desenvolvimento de software.

3 - Existem quatro atividades fundamentais comuns a todos os processos de software. Quais são?

Resposta: Especificação, Desenvolvimento, Validade e Evolução de Software.

APÊNDICE 24 – Pergunta das Cartas - Tema: Engenharia de Requisitos

CARTA 1

1 - A prototipagem, técnica que serve de complemento à análise de requisitos, é a construção de protótipos considerados um esboço de parte do sistema.

a) Verdadeiro b) Falso

2 - As principais técnicas de elicitação de requisitos podem ser classificadas em quatro categorias: tradicionais, colaborativas, cognitivas e abordagens contextuais. Na categoria colaborativa encontram-se:

- a) análise de tarefas, análise de protocolos e reuniões.
- b) entrevistas, brainstorms e reuniões tecnossociais.
- c) entrevistas, pesquisas e questionários, reuniões e cenários.

d) grupo focal, brainstorms e prototipação.

3 - Identificar o conhecimento e os requisitos dos *stakeholders* é uma atividade importante do ciclo de desenvolvimento de um projeto de software. Considere que você tem a responsabilidade de desenvolver requisitos inovadores, porém, neste momento não está muito interessado em estabelecer requisitos precisos sobre o comportamento do sistema. Qual é a técnica de elicitação de requisitos mais indicada para esboçar uma visão inicial do sistema e identificar fatores inesperados de satisfação dos *stakeholders*?

Resposta: Brainstorm

CARTA 2

1 - No que se refere à engenharia de software, aos modelos de ciclo de vida e aos requisitos funcionais e não funcionais, julgue o item a seguir.

Os requisitos não funcionais utilizam, para especificação, métricas como velocidade, confiabilidade, tamanho, análise de requisitos e classificação dos requisitos.

a) Verdadeiro **b) Falso**

2 - A elicitação de requisitos deve ser realizada em etapas. Dentre as diversas formas de estabelecimento dessas etapas, há uma que considera 4 etapas, cuja ordem sequencial correta de realização é:

a) Especificação dos Requisitos, Priorização e Negociação dos Requisitos, Descoberta dos Requisitos, Classificação e Organização dos Requisitos.

b) Classificação e Organização dos Requisitos, Descoberta dos Requisitos, Especificação dos Requisitos, Priorização e Negociação dos Requisitos.

c) Descoberta dos Requisitos, Classificação e Organização dos Requisitos, Priorização e Negociação dos Requisitos, Especificação dos Requisitos.

d) Priorização e Negociação dos Requisitos, Especificação dos Requisitos, Classificação e Organização dos Requisitos, Descoberta dos Requisitos.

3 - O gerenciamento de requisitos constitui-se em uma fase importante na engenharia de requisitos. Nesse gerenciamento, deve haver o acompanhamento e o conhecimento da origem dos requisitos do sistema, o que corresponde à propriedade de:

Resposta: rastreabilidade.

CARTA 3

1 - Engenharia de requisitos compreende: identificar, analisar, especificar e definir as necessidades de negócio que um aplicativo deve prover para solução do problema levantado.

a) **Verdadeiro** b) Falso

2 - O gerenciamento de requisitos constitui-se em uma fase importante na engenharia de requisitos. Nesse gerenciamento, deve haver o acompanhamento e o conhecimento da origem dos requisitos do sistema, o que corresponde à propriedade de:

a) atomicidade.

b) durabilidade.

c) integridade.

d) rastreabilidade.

3 - As políticas de rastreabilidade de requisitos são decididas durante qual estagio?

Resposta: gerenciamento de requisitos.

CARTA 4

1 - O principal artefato elaborado no processo de produção de requisitos do sistema, segundo a Engenharia de Requisitos, é o documento de requisitos. Por sua vez, o documento de requisitos é uma declaração formal dos requisitos para os stakeholders, que podem ser clientes, usuários finais ou a equipe de desenvolvimento do software.

a) **Verdadeiro** b) Falso

2 - Ao determinar os requisitos de um projeto, é necessário levantar os requisitos funcionais e os não funcionais. Os requisitos não funcionais:

a) definem com detalhes exatamente o que deve ser implementado.

b) definem explicitamente as funções que o sistema não deve executar.

c) **representam restrições aos serviços oferecidos pelo sistema.**

d) indicam os serviços que o sistema deve prestar.

3 - Os requisitos não funcionais não estão diretamente ligados aos serviços específicos oferecidos pelo sistema a seus usuários. Eles podem estar relacionados às propriedades emergentes do sistema, como confiabilidade, tempo de resposta e ocupação de área, entre outros. Dentre os tipos de requisitos não funcionais, é possível destacar os requisitos de produto, organizacionais e externos. Dentre os requisitos de produto, podemos citar os requisitos de?

Resposta: eficiência e de confiança.

CARTA 5

1 - São consideradas técnicas de validação de requisitos: revisões de requisitos, prototipação e geração de casos de teste.

a) **Verdadeiro** b) Falso

2 - Na engenharia de requisitos, a etapa em que a equipe de revisão examina a especificação do sistema, procurando erros de conteúdo ou interpretação, áreas em que esclarecimentos podem ser necessários, informação omissa, inconsistências, requisitos conflitantes ou requisitos não realísticos, é conhecida como:

a) análise e negociação de requisitos;

b) modelagem do sistema;

c) especificação de requisitos;

d) validação de requisitos.

3 - Na validação de requisitos, parte integrante da especificação desses requisitos, é correto o uso de diagramas da UML, quais são?

Resposta: diagrama de classes, de casos de uso e de interação.

CARTA 6

1 - Requisitos não funcionais são restrições sobre os serviços ou as funções oferecidas pelo sistema, e podem ser também, declarações de serviços que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como deve comportar-se em diversas situações.

a) Verdadeiro **b) Falso**

2 - É considerado um requisito NÃO funcional

a) o saldo calculado.

b) o tempo de resposta máximo.

c) o ponto de reposição.

d) a requisição baixada.

3 - Considere o seguinte texto extraído de um documento de requisitos:

“Os dados devem ser armazenados no banco de dados utilizando um formato padrão e hierárquico, que facilite sua leitura e a geração de relatórios.” Qual exemplo de requisito o texto apresenta?

Resposta: Não funcional

CARTA 7

1 - A documentação de requisitos deve conter duas perspectivas: uma voltada para o cliente, em linguagem compreensível por ele, e outra voltada para o desenvolvedor, em uma linguagem técnica de modelagem.

a) **Verdadeiro** b) Falso

2 - No processo de engenharia de requisitos, os tipos de requisitos de *usuário* e de *sistema* podem ser, respectivamente,

- a) apenas funcionais; apenas não funcionais.
- b) apenas não funcionais; apenas funcionais.
- c) apenas funcionais; funcionais e não funcionais.

d) funcionais e não funcionais; funcionais e não funcionais.

3 - No processo de engenharia de software, os requisitos funcionais podem ser também definidos como requisitos de?

Resposta: Capacidade

CARTA 8

1 - O gerenciamento de mudanças de requisitos de software se propõe a garantir o escopo de um sistema a partir da formalização de procedimentos entre os envolvidos no desenvolvimento para a solicitação, a avaliação, a aprovação e a garantia da implementação das mudanças nos requisitos.

a) **Verdadeiro** b) Falso

2 - De acordo com Sommerville, são atividades do processo de elicitação de requisitos, pela ordem:

- a) casos de uso; análise; projeto; arquitetura.
- b) obtenção; classificação e organização; priorização e negociação; documentação.**
- c) entrevista; etnografia; documentação; registro.
- d) cenários; classificação; organização; priorização; documentação.

3 - Quais são requisitos relativamente estáveis derivados da atividade central da organização e que se relacionam diretamente ao domínio do sistema. **Resposta: Requisitos permanentes**

CARTA 9

1 - Assim como o software, os requisitos também devem ser avaliados quanto à qualidade. A validação, atividade da engenharia de requisitos, é responsável por garantir que os requisitos tenham sido declarados de forma clara e precisa. Além disso, a validação busca detectar inconsistências, erros e omissões, objetivando alinhar os requisitos às normas estabelecidas para o projeto, produto e processo.

a) **Verdadeiro** b) Falso

2 - Entre as atividades listadas a seguir, uma não faz parte da Engenharia de Requisitos.

- a) estudo de viabilidade.
- b) **análise de risco.**
- c) levantamento de necessidades do cliente.
- d) verificação e gerenciamento.

3 - São do tipo não funcionais, relacionado ao produto software, APENAS os requisitos de?

Resposta: desempenho e de portabilidade.

APÊNDICE 25 – Pergunta das Cartas – Tema: Modelos Ágeis

CARTA 1

1 - O XP (extreme programming) é um método ágil, que preconiza a criação de um caso de teste unitário antes do início da codificação.

a) **Verdadeiro** b) falso

2 - Sobre XP e SCRUM é INCORRETO afirmar:

- a) No XP, os testes são escritos antes da atividade de desenvolvimento e todas as funcionalidades só possuem valor se forem testadas e obtiverem unanimidade de aprovação.
- b) No XP, não há indicação de que é necessário criar documentação no código porém, os documentos tradicionais são reduzidos aos aspectos mais relevantes, visando obter no final do processo, apenas artefatos de grande importância para o projeto.

c) SCRUM não especifica a programação em pares ou desenvolvimento orientado a testes, porém especifica a forma de gerenciamento dos requisitos ou características solicitadas.

d) Enquanto o XP é mais receptivo a mudanças durante a iteração, no SCRUM as solicitações do cliente devem aguardar o término da iteração em andamento.

3 - Em reunião, toda conversação é restringida às respostas dos elementos às perguntas colocadas pelo *Scrum Master*, sendo uma delas: "O que planeja desenvolver até a próxima reunião?" As *Scrum meetings* ocorrem?

Resposta: Diariamente.

CARTA 2

1 - Scrum é um processo de desenvolvimento que tem como ponto de partida um conjunto de requisitos bem definidos.

- a) Verdadeira **b) Falsa**

2 - Nos métodos ágeis XP e Scrum, as entregas de partes funcionais do projeto são divididas em ciclos, geralmente compreendidos no período de 1 a 4 semanas. Estes ciclos denominam-se, respectivamente,

a) iterações e sprint.

- b) reunião de planejamento e backlog.
c) período de entrega e reunião de revisão.
d) backlog e planejamento da produção.

3 - No SCRUM, que papel é responsável pela visão do produto e pelo retorno do investimento?

Resposta: Product Owner

CARTA 3

1 - Uma das características do método XP é o uso de um modo de desenvolvimento orientado a testes frequentes, o que garante a entrega de uma única versão do sistema inteiro, testado e validado.

- a) Verdadeiro **b) falso**

2 - Assegurar que a equipe se concentre em fazer, primeiro, apenas aquilo que é claramente necessário e evite fazer o que poderia vir a ser necessário, mas ainda não se provou essencial. Este é um dos cinco valores fundamentais do XP (Extreme Programming), denominado

- a) coragem
b) **Simplicidade**
c) comunicação
d) feedback

3 - Refactoring, Programação Em Pares E Stand-Up Meeting são características das práticas do?

Resposta: XP - Extreme Programming.

CARTA 4

1 - O desenvolvimento ágil de sistemas consiste em uma linguagem de modelagem que permite aos desenvolvedores visualizarem os produtos de seu trabalho em gráficos padronizados.

- a) Verdadeira **b) Falsa**

2 - São algumas das metodologias de desenvolvimento de software consideradas ágeis (Agile Software Process Models)

a) RUP, XP e DSDM.

b) Scrum, XP e FDD.

c) Waterfall, RUP e FDD.

d) XP, Scrum e RUP.

3 - Na metodologia de desenvolvimento ágil, a prática da programação em par sugere que todo e qualquer código produzido no projeto seja sempre implementado por duas pessoas juntas. Como é denominado o papel da pessoa que revê cada linha de código enquanto ela é digitada, verificando erros e pensando sobre o projeto global? **Resposta: Navegador**

CARTA 5

1 - Acerca de engenharia de software, que permite a criação, de maneira econômica e confiável, de software que trabalhe eficientemente em máquinas reais, julgue os próximos itens. Para que se obtenha sucesso na utilização do Scrum, o cliente deve se tornar parte da equipe de desenvolvimento do software, participando diretamente do processo.

a) Verdadeiro b) falso

2 – No SCRUM, o produto final, a data final e o custo do projeto são determinados:

a) respectivamente, no planejamento, ao longo do projeto, no início do projeto.

b) ao longo do projeto.

c) no planejamento.

d) respectivamente, nas fases intermediárias, no planejamento, no final do projeto.

3 - Os princípios Scrum são usados para guiar as atividades de desenvolvimento dentro de um processo que incorpora as seguintes atividades de arcabouço: requisitos, análise, projeto, evolução e entrega. Em cada atividade de arcabouço, as tarefas de trabalho ocorrem dentro de um padrão de processo chamado de?

Resposta: sprint.

CARTA 6

1 - Desenvolvimento ágil de software ou método ágil é aplicado, principalmente, a grandes corporações, uma vez que permite produzir grandes sistemas de forma ágil.

a) Verdadeira **b) Falsa**

2 - É uma premissa típica do desenvolvimento ágil:

a) O cliente aprende ao longo do desenvolvimento, à medida que é capaz de manipular

o sistema.

b) A especialização torna as tarefas mais simples e consequentemente facilita o determinismo.

c) O foco na execução torna o processo determinístico, para que a especificação seja transformada corretamente em software.

d) Busca por processos determinísticos, visando menos alterações e maior previsibilidade.

3 - Trata-se de um modo comum de aplicar a UML, frequentemente com alto retorno no investimento de tempo. Essa definição refere-se a:

Resposta: Modelagem ágil.

CARTA 7

1 - Na extreme programming, os requisitos são expressos como cenários e implementados diretamente como uma série de tarefas. O representante do cliente faz parte do desenvolvimento e é responsável pela definição de testes de aceitação do sistema.

a) Verdadeiro b) falso

2 - São práticas recomendadas pelo processo ágil de desenvolvimento de software Extreme Programming (XP), EXCETO a

a) Programação em Pares

b) Documentação Abundante e Detalhada

c) Integração Contínua

d) Refatoração Frequente

3 - A Extreme Programming (XP) baseia-se em 12 práticas, que são um conjunto de atividades que deverão ser seguidas pelas equipes que desejam utilizar a XP. Na prática do Jogo do Planejamento, as funcionalidades são descritas em pequenos cartões que são conhecidos como?

Respostas: histórias de usuário.

APÊNDICE 26 – Pergunta das Cartas – Tema: Processo unificado

CARTA 1

1 - O processo unificado de desenvolvimento de software é o conjunto de atividades necessárias para transformar requisitos do usuário em um sistema de software, baseado em componentes.

a) **Verdadeiro** b) falso

2 - Em ordem cronológica, quais são as fases definidas no Processo Unificado (UP)?

- a) Concepção, Requisitos, Implementação e Testes
- b) Concepção, Elaboração, Construção e Implantação
- c) **Concepção, Elaboração, Construção e Transição**
- d) Elaboração, Concepção, Construção e Transição

3 - No Processo Unificado, os casos de uso mais importantes são capturados e delimitam o domínio do sistema durante a fase de? **Resposta: Concepção**

CARTA 2

1- No Processo Unificado, a elicitação de requisitos do sistema de software inicia se na fase de concepção.

a) **Verdadeiro** b) falso

2 - No Processo Unificado os papéis não são pessoas; eles descrevem como as pessoas se comportam no negócio e quais são as responsabilidades que elas têm. Nesse sentido, um Desenvolvedor desempenha os papéis de:

- a) Analista de Sistemas, Designer de Negócios e Revisor de Requisitos.
- b) Revisor de Requisitos, Analista de Teste e Revisor do Projeto.
- c) Implementador, Integrador e Designer.
- d) **Gerente de Teste, Gerente de Projeto e Designer.**

3 - No Processo Unificado (UP), que nome é dado à diferença (delta) entre dois releases do produto ao final de iterações subsequentes? **Resposta: Incremento**

CARTA 3

1 - O processo unificado de software é centrado na arquitetura e orientado por casos de uso, o que sugere um fluxo de processo iterativo e incremental.

a) **Verdadeiro** b) falso

2 - No Processo Unificado de desenvolvimento de software, Requisitos e Testes são?

a) designações das fases do ciclo de vida.

b) designações dos fluxos de trabalho.

c) disciplinas consideradas apenas na quarta fase.

d) disciplinas consideradas apenas na segunda fase.

3 - O Processo Unificado foi estabelecido em termos de fases que são: Concepção, Elaboração, Construção, Transição e Produção. Uma variedade de produtos de trabalho pode ser produzida como consequência da aplicação da UML. Um produto de trabalho da fase de Concepção é o: **Resposta: modelo inicial de caso de uso**

CARTA 4

1 – Dirigido por caso de uso, orientado por quatro workflows, iterativo e incremental são características do Processo Unificado?

a) **Verdadeiro** b) falso

2 - São fluxos de trabalho do Processo Unificado (UP), **EXCETO:**

a) Análise.

b) Construção.

c) Desenho.

d) Requisitos.

3 - No UP, a fase que cobre o período em que o produto fica em versão beta é a de:

Resposta: Transição

CARTA 5

1 - No Processo Unificado, modelos de projeto contêm classes de projeto e seus objetos; cada classe de projeto representa uma abstração de classes no modelo de análise; em uma classe de projeto, os atributos são conceituais e os comportamentos são definidos via responsabilidades, que são descrições textuais dos comportamentos.

a) Verdadeiro **b) falso**

2 - O Processo Unificado é dividido em diversas fases, e em cada uma delas podem ser realizadas atividades de diferentes fluxos de trabalho (workflows), em diferentes proporções. A característica

que **NÃO** se aplica a esse processo é ser

- a) incremental.
- b) iterativo.
- c) centrado em arquitetura.
- d) modelo prescritivo.
- e) guiado por testes de aceitação.**

3 - No Processo Unificado (UP), o fluxo de trabalho Análise, tem forte concentração na fase de?

Resposta: Elaboração

APÊNDICE 27 – Pergunta das Cartas - Tema: Processos de Software

CARTA 1

1- Os estágios do processo de software incluem situação atual, definição do problema, desenvolvimento técnico e integração da solução.

a) Verdadeiro b) falso

2 - Um processo de software é um conjunto de atividades relacionadas que levam à produção de um produto de software. Existem muitos processos de software diferentes, mas todos devem incluir quatro atividades fundamentais: especificação, projeto e implementação, validação e:

- a) teste
- b) evolução.**
- c) prototipação.
- d) entrega.

3- FDD (Feature Driven Development) é uma metodologia muito objetiva, possuindo apenas duas fases, quais? **Resposta: Concepção e Planejamento e Construção.**

CARTA 2

1 - Podemos definir um processo de software como um conjunto de atividades relacionadas que levam à produção de um produto de software.

a) Verdadeiro b) falso

2 - O modelo de processo de desenvolvimento de software que combina as atividades de desenvolvimento com o gerenciamento de risco é o Modelo?

a) Transformacional

b) em Espiral

c) em V

d) em Cascata

3 - O conjunto de atividades e resultados associados que resulta em um produto de software recebe o nome de? **Resposta: Processo de Software**

CARTA 3

1 - São elementos de um processo de desenvolvimento de *software*: atividade, sequência, modelo de processo, recursos, controles, políticas e organização.

a) Verdadeiro b) falso

2 - Metodologias de desenvolvimento de software se baseiam em um modelo de ciclo de vida, tais como cascata, espiral e prototipagem; sendo assim, é correto afirmar que:

a) metodologias que seguem o modelo em espiral normalmente possuem um maior potencial de risco, uma vez que esse modelo não lida explicitamente com isso.

b) metodologias que seguem o modelo de prototipagem devem, necessariamente, descartar os protótipos construídos; dessa forma, essas metodologias costumam ser mais custosas.

c) metodologias que seguem o modelo em cascata possuem fases bem definidas e executadas sequencialmente. Além disso, não há sobreposição entre as fases, isto é, a fase seguinte somente pode ser executada após a finalização da fase anterior.

d) em metodologias que seguem o modelo em espiral, o software é desenvolvido em apenas uma iteração.

3 – Que paradigma da Engenharia de Software é sequencial e sistemático, iniciando no nível de sistemas e se estendendo pela análise, projeto, codificação, teste e manutenção?

Resposta: Clássico

CARTA 4

1 - Um modelo de processo de software descreve os processos que são realizados para atingir o seu desenvolvimento. A notação para as tarefas, os artefatos, os atores e as decisões varia conforme o modelo de processo utilizado.

a) Verdadeiro b) falso

2 - Em relação à engenharia de software, a afirmação **NÃO** pertinente aos processos de software é:

- a) são atividades ordenadas, normalmente divididos em fases;
- b) usam uma sistematização normalmente rígida e pesada;
- c) suas fases buscam manter a integridade e a qualidade do produto final;

d) um dos primeiros processos de software existente foi o modelo espiral.

3-As atividades fundamentais relacionadas ao processo de construção de um software incluem a especificação, o desenvolvimento, a validação e a?

Resposta: evolução do software.

CARTA 5

1 – A definição das funcionalidades do software e as restrições a seu funcionamento devem ser definidas na produção de um software. Essa atividade está incluída no processo de software.

a) Verdadeiro b) falso

2 - Em cada fase de um processo de software são executadas as atividades básicas para que sejam atingidos os objetivos propostos. Essas atividades podem ser identificadas nas alternativas a seguir, à exceção de uma.

a) Projeto.

b) Integração.

c) Especificação.

d) Implementação.

3 - O software possui um ciclo de vida. Cada fase do ciclo de vida possui divisões e subdivisões. Em qual fase avaliamos a necessidade de evolução dos softwares em funcionamento para novas plataformas operacionais ou para a incorporação de novos requisitos?

Resposta: Fase de retirada;

APÊNDICE 28 – Pergunta das Cartas - Tema: Modelos de Processos Prescritivos

CARTA 1

1 - No desenvolvimento em cascata, os passos são ordenados em sequência e executados até a etapa final de teste, mesmo que o cliente esteja insatisfeito com determinado resultado intermediário.

a) Verdadeiro b) falso

2 - A principal metodologia tradicional utilizada no desenvolvimento de software é o modelo clássico também conhecido como cascata ou sequencial. Nesse modelo,

a) cada etapa tem associada ao seu término uma documentação que deve ser aprovada para que a etapa posterior possa ter início.

b) o projeto é dividido em fases de maneira flexível.

c) utiliza-se o desenvolvimento incremental e iterativo.

d) os requisitos não podem ser estáveis.

3 – O modelo de ciclo de vida de processo de software cujos principais subprocessos são executados em estrita sequência, o que permite demarcá-los como pontos de controle bem definidos, é denominado: **Resposta: Cascata**

CARTA 2

1 - A realização de protótipos descartáveis, com o objetivo de compreender os requisitos do cliente e desenvolver melhor definição do produto, faz parte do modelo de desenvolvimento em cascata.

a) Verdadeiro **b) falso**

2 - O modelo em cascata inclui 5 estágios considerados fundamentais para o desenvolvimento de um software: a análise e definição de requisitos, o projeto de sistema e software, a implementação e o teste de unidade, a integração e o teste de sistema e a operação e manutenção. Apesar disso, o modelo em cascata tem como desvantagem a:

a) documentação produzida em cada estágio.

b) aderência a outros modelos de processo de engenharia.

c) dificuldade de reação a mudanças de requisitos do usuário.

d) falta de estruturação para desenvolvimento de software.

3 - O modelo espiral para a Engenharia de Software foi desenvolvido acrescentando-se novos elementos às melhores características de outros modelos. Segundo o modelo espiral, a determinação dos objetivos, alternativas e restrições está relacionada à atividade de?

Resposta: planejamento.

CARTA 3

1 - O modelo em espiral é um modelo de processos de software que reúne a natureza iterativa da prototipação com os aspectos sistemáticos e controlados do modelo sequencial linear.

a) Verdadeiro b) falso

2 - No modelo de desenvolvimento em espiral, cada ciclo da espiral representa uma fase do processo de software. Nesse modelo, a atividade que obrigatoriamente estará presente em todos os ciclos é:

a) Análise de riscos.

b) Análise de requisitos.

c) Teste de unidade.

d) Análise, Projeto, Implementação e Teste.

3 – O ciclo de vida de um software pode ser descrito com base em modelos. Um deles, proposto por Barry Boehm em 1988, apresenta-se como um modelo onde cada fase é precedida por uma análise de risco e sua execução é realizada incrementalmente. O nome dado ao modelo descrito por Barry Boehm em 1988 é modelo?

Resposta: Espiral

CARTA 4

1 – No Desenvolvimento em Espiral, cada loop da espiral é dividido em quatro setores, Desenvolvimento de integração, Definição de objetivos, Avaliação e redução de riscos e Planejamento.

a) Verdadeiro b) falso

2- No modelo de desenvolvimento em espiral, cada *loop* da espiral representa uma fase do processo de software. A importante distinção entre este modelo e os demais é a consideração em todos os ciclos da análise de:

a) escopo.

b) requisitos.

c) implementação e teste.

d) riscos.

3 - No modelo em espiral do processo de software cada loop na espiral representa?

Resposta: Uma fase do processo.

CARTA 5

1 - Ciclo de vida de um software resume-se em eventos utilizados para definir o status de um projeto.

a) Verdadeiro **b) falso**

2 - O modelo de ciclo de vida em cascata:

a) enfatiza a realização sequencial das atividades do desenvolvimento de um produto de software.

b) enfatiza a comunicação estreita com o cliente durante o desenvolvimento do produto de software.

c) envolve a análise dos riscos envolvidos no desenvolvimento dos requisitos identificados para produto de software.

d) recomenda a geração de versões incompletas do sistema, que podem ser passadas para o usuário final, o que permite a retroalimentação do processo de desenvolvimento.

3 - O processo de engenharia de software denominado ciclo de vida clássico refere-se ao modelo?

Resposta: Cascata

APÊNDICE 29 – Pergunta das Cartas - Tema: Gerenciamento de Configuração e Mudança

CARTA 1

1 – Integração contínua: tem como características testar as mudanças, assim que são realizadas.

- a) **Verdadeira** b) Falso

2 – Na disciplina de Gerência de Configuração e Mudança, o artefato repositório do projeto:

- a) pode conter compilações, builds ou teste.
b) pode executar ferramentas de terceiros.
c) pode compartilhar o hardware com um servidor Web.

d) deve estar em um servidor exclusivo para armazenar as versões de diretórios e arquivos do projeto

3 – O Gerenciamento de Configuração de Software é dividido em três níveis. Quais são? **Resposta: Controle de versão, Controle de mudanças e Integração contínua.**

CARTA 2

1 – Controle de mudanças: identifica quais foram as mudanças, efetuadas na versão.

- a) **Verdadeira** b) Falso

2 – Quando um sistema de software passa por manutenção, deve-se assegurar que as mudanças incorporadas a esse software sejam controladas. A esse processo de gerenciamento de mudanças do sistema dá-se o nome de gerenciamento de configuração de software, que inclui as atividades de:

- a) projeto detalhado e teste de software.
b) teste de software e gerenciamento de versões.

c) gerenciamento de versões e construção de sistemas.

d) especificação de requisitos e garantia de qualidade de software.

3 – Qual é o conceito de Gerenciamento de Configuração de Software que ajuda a controlar as mudanças sem impedir seriamente as mudanças justificáveis? **Resposta: Linha de Referência (baseline).**

CARTA 3

1 – Controle de versão: identifica em qual versão está o software e quais as características de cada versão.

b) Verdadeira b) Falso

2 – O processo de Gerência de Configuração segundo a norma ISO 12207, apresenta várias etapas. A seguir estão relacionadas algumas dessas etapas, à **EXCEÇÃO** de uma. Assinale-a.

a) Implementação do produto.

b) Implementação do processo.

c) Identificação da configuração.

d) Relato da situação da configuração.

3 – Um local sob controle de acesso (banco de dados) onde são armazenados os Itens de Configuração de Software depois de liberados por uma "baseline" é chamado de? **Resposta: Repositório de Itens de Configuração**

CARTA 4

1 – Na gestão de configuração de software, a manutenção pode ser feita de maneira proativa, reativa, preventiva ou corretiva.

a) Verdadeira **b) Falso**

2 – O processo de Gerência de Configuração de Software é definido por quatro funções básicas:

a) Armazenagem, Utilização, Alteração e Personalização.

b) Classificação, Agrupamento, Utilização e Manipulação.

c) Identificação, Documentação, Controle e Auditoria.

d) Públicas, Privadas, Atribuídas e Herdadas.

3 – O método utilizado para trabalhar com itens de configuração que já estão no repositório, ou seja, conferência na entrada e conferência na saída é chamado de? **Resposta: Check In/Check Out**

CARTA 5

1 – Os sistemas de controle de versões permitem que os artefatos sejam obtidos do repositório por meio da operação conhecida como check-in.

a) Verdadeira **b) Falso**

2 – Sistemas de controles de versões são ferramentas essenciais na gestão de tecnologia da informação de empresas, em especial em empresas desenvolvedoras de software. Estes sistemas têm o intuito de:

a) controlar as versões dos diversos softwares adquiridos pela empresa.

- b) calcular as funcionalidades do sistema, incluindo cálculos de pontos de função.
- c) identificar uma alteração específica efetuada em um código fonte.
- d) estimar o custo e tempo de desenvolvimento de uma versão específica de um sistema.

3 – Uma ferramenta de controle de versão é utilizada para controlar as diferentes versões de um documento, o qual pode possuir ramificações de sua linha de desenvolvimento original. Um termo técnico comum, utilizado para definir uma ramificação é? **Resposta: Branch.**

CARTA 6

1 - Gerência de Configuração de Software é um conjunto de atividades de apoio que permite a absorção ordenada das mudanças inerentes ao desenvolvimento de software, mantendo a integridade e a estabilidade durante a evolução do projeto.

- a) **Verdadeiro** b) Falso

2 – Qual dos itens a seguir **NÃO** são Tarefas de Gerenciamento de Configuração:

- a) Tarefas Preliminares e Auditoria de Configuração
- b) Identificação
- c) Controle de Mudanças e Controle de Versão

d) Controle de Segurança

3 – Assegurar que as mudanças em um software sejam feitas de modo controlado, permitindo-se prever o efeito das mesmas em todo o sistema, é responsabilidade do? **Resposta: Controle de Mudança.**

CARTA 7

1 - O Gerenciamento de Configuração do Software é um importante elemento da garantia da qualidade de software.

- a) **Verdadeiro** b) Falso

2 – Sobre as tarefas de Gerenciamento de Configuração - Tarefas Preliminares geralmente, não devem sofrer gerenciamento de configuração:

- a) Os itens mais usados no ciclo de vida
- b) **Os itens menos genéricos**
- c) Os itens mais importantes para a segurança
- d) Os itens projetados para reuso

3 - Um item, ao ser desenvolvido, evolui até que atinja um estado em que atenda aos propósitos para

o qual foi criado. Isso implica em diversas alterações, gerando uma versão do item a cada estado, sendo necessário a utilização de um controle de? **Resposta: Versões.**

APÊNDICE 30 – Pergunta das Cartas - Tema: Gerenciamento de projeto de software

CARTA 1

1 - No contexto da gerência do projeto, integração inclui características como unificação, consolidação, articulação e ações de integração que são cruciais para concluir o projeto, atender satisfatoriamente os requisitos dos interessados e clientes e gerenciar as expectativas.

c) **Verdadeiro** b) Falso

2 - Sobre gerência de projetos, é **INCORRETO** afirmar que:

a) não existe necessariamente uma relação direta entre o valor cobrado por um sistema e o custo de seu desenvolvimento.

b) o tempo necessário para a finalização de um projeto é sempre proporcional ao número de pessoas alocadas a ele

c) entre os fatores que alteram a produtividade de um software estão o ambiente de trabalho e o processo de desenvolvimento.

d) existem diversas técnicas de estimativa de custos de sistema e, ao se preparar a estimativa de custo, diversas técnicas devem ser utilizadas.

3 – Como é denominado o conjunto de fases coletivas atravessadas pelo projeto? **Resposta: ciclo de vida do projeto.**

CARTA 2

1 - No contexto dos requisitos e restrições de um projeto, o propósito da gerência de projetos é identificar, estabelecer, coordenar e monitorar as atividades, tarefas e recursos de que um projeto necessita para produzir um produto.

a) **Verdadeira** b) Falso

2 - O ponto de partida para o planejamento do projeto é o:

a) Plano de trabalho.

b) Descritivo das funcionalidades.

c) Escopo

d) Documento de autorização do projeto.

3 - Em que grupo de processos de gerenciamento de um projeto o gerente deve envolver as partes interessadas para aumentar a probabilidade de que elas fiquem satisfeitas e aceitem as entregas do projeto? **Resposta: Iniciação**

CARTA 3

1 – Na área de projeto de software, também conhecida como design de software, o software começa a ser implementado e validado pelos programadores.

b) Verdadeira **b) Falso**

2 – Com Ferramentas de Gerenciamento de Projetos, o gerente de projeto pode definir uma:

a) equipe de definição do trabalho.

b) estrutura de apoio ao usuário.

c) estrutura de divisão do trabalho.

d) estrutura de divisão de objetos.

3 – No processo de gerenciamento de configuração de software, um conjunto de itens de configuração que deve ser controlado, formalmente designado e fixado num tempo específico do ciclo de vida do software, é denominado? **Resposta: Baseline.**

CARTA 4

1 – O plano de gerencia de requisitos apresenta todas as atividades de um projeto.

a) Verdadeira **b) Falso**

2 – O gerente de um projeto de software identificou atraso em seu cronograma. O que ele pode fazer para que os prazos desse cronograma sejam cumpridos?

a) Documentar os procedimentos de solicitações de mudanças no escopo.

b) Alterar o escopo, removendo funcionalidades complexas do sistema.

c) Retirar recursos especializados das tarefas do caminho crítico do projeto.

d) Adicionar recursos especializados às tarefas do caminho crítico do projeto.

3 – Ciclo de vida de um projeto compõe-se de quais fases? **Resposta: iniciação, planejamento, execução e finalização.**

CARTA 5

1 – Na gerencia de projetos qualidade e cronograma podem ser definidos independentemente.

a) **Verdadeira** b) Falso

2 – No processo Conceber Novo Projeto da disciplina Gerenciamento de Projeto, a atividade Iniciar Projeto é realizada após a aprovação do artefato:

a) **caso de negócio.**

b) plano de interação.

c) visão.

d) lista de risco.

3 – O Projeto de software é o primeiro passo da fase de desenvolvimento de qualquer produto ou sistema de engenharia. Do ponto de vista técnico, a fase de projeto produz: **Resposta: projeto de dados, arquitetural, procedimental e interface.**

APÊNDICE 31 – Pergunta das Cartas - Tema: Planejamento de Software

CARTA 1

1 – O planejamento de software é uma tarefa importante e inclui Identificar tarefas e alocá-las aos membros da equipe Antecipar problemas e preparar soluções.

a) **Verdadeiro** b) Falso

2 -Todos os processos abaixo fazem parte do grupo de processos de planejamento, exceto qual?

a) Coletar os Requisitos

b) Determinar o Orçamento

c) **Conduzir as Aquisições**

d) Realizar a Análise Quantitativa dos Riscos

3 – O planejamento é monitorado e modificado periodicamente em qual fase?

Resposta: Fase de Atualização

CARTA 2

1 – planejamento ocorre em três fases: Fase da proposta; Fase de iniciação do projeto e Fase de atualização do planejamento.

a) **Verdadeiro** b) Falso

2 – Qual das opções abaixo **NÃO** são atividades do Planejamento de software:

- a) Elaboração de propostas
- b) Planejamento e cronograma de projeto
- c) **Testes e Releases**
- d) Monitoramento e revisões

3 – Elabora um contrato, decidir se há recursos suficientes e Estimar o preço a ser cobrado são requisitos da Fase de? **Resposta: Fase da Proposta.**

CARTA 3

1 - Tradicionalmente, durante o planejamento de desenvolvimento de software, são definidos prazos, custos, recursos e o escopo do projeto.

A) **Verdadeiro** b) Falso

2 - A Fase de Planejamento e Elaboração **NÃO** inclui:

- a) Criar relatório inicial de investigação (para construir o business case)
- b) Levantar requisitos funcionais e não funcionais
- c) **Atribuição de responsabilidades entre os objetos**
- d) Definir modelo conceitual inicial (análise inicial) e Projetar arquitetura

3 - O planejamento ocorre em três fases, quais são?

Resposta: Fase da proposta, fase de iniciação do projeto e fase de atualização do planejamento.

CARTA 4

1 – No Planejamento do Projeto de Software devem ser derivados: estimativa do esforço humano exigido, duração cronológica e custo.

a) **Verdadeiro** b) Falso

2 – Qual das alternativas abaixo **NÃO** são atividades fundamentais do planejamento de projeto:

- a) **Controle de custos.**
- b) Elaboração de Estimativas;
- c) Análise de Riscos;
- d) Elaboração de Cronograma;

3 – No planejamento é realizada a estimativa de custos, quais os três parâmetros principais para estimar o preço total do Software? **Resposta: Custos de esforço, Custos de hardware e software, Custos de viagens e treinamentos.**

APÊNDICE 32 – Pergunta das Cartas - Tema: Riscos

CARTA 1

1 - A projeção dos riscos, também chamada estimativa dos riscos, tenta classificar cada risco de duas maneiras – a probabilidade de que o risco seja real e as consequências dos problemas associados ao risco, caso ele ocorra.

- a) **Verdadeiro** b) Falso

2 - As estratégias de gerenciamento de risco são divididas em três categorias. Uma dessas categorias é a estratégia:

a) organizacional.

b) de minimização

c) de análise de riscos.

d) de monitoramento de riscos

3 – Qual é modelo que visa a melhoria de processos que fornece às organizações elementos essenciais de processos eficazes e pode ser usado para guiar a melhoria de processo em um projeto, divisão ou em uma organização inteira. **Resposta: CMMI**

CARTA 2

1 – São atividades da Gerência de Risco, Planejar a gerência de risco, Identificar riscos, Analisar riscos, Planejar respostas aos riscos, Monitorar riscos, Controlar riscos e Comunicar os riscos.

- a) **Verdadeiro** b) Falso

2 - Sobre gerenciamento de riscos de software, assinale a alternativa correta:

a) O gerenciamento de riscos de software não envolve os riscos ambientais, que possam afetar o projeto.

b) O controle de riscos é feito pelos planos de casos de uso

c) O controle de riscos é realizado por membros externos ao projeto.

d) Gerenciamento de riscos de software consiste em avaliar e controlar os riscos, que afetam o projeto, processo ou produto de software.

3 - Associadas à especificação de sistemas críticos, as técnicas de decomposição de riscos podem ser (1) dedutivas - do risco em direção à falha possível ou, ao contrário, (2) indutivas - da falha proposta em direção aos possíveis perigos que levariam a ocorrer, ou seja, respectivamente, técnicas? **Top-**

down e bottom-up.**CARTA 3**

1 – A atividade de planejar a gerência de risco tem a finalidade de definir a estratégia de gerenciamento de risco, dos recursos necessários para a realização do processo e por fim, da efetivação das ações consideradas necessárias no plano de gerência de risco.

a) Verdadeiro b) Falso

2 - Na especificação dirigida a riscos, a compreensão da probabilidade de ocorrência de um risco e das consequências potenciais, se um acidente ou incidente, associado com este risco, ocorrer, é da competência do processo de:

- a) especificação de requisitos.
- b) análise da qualidade.
- c) validação de requisitos.

d) análise e classificação de riscos.

3 – Em que atividade se avalia a situação corrente para determinar eventuais desvios do planejado e envolve alteração das estratégias de mitigação, quando se fizer necessário; utilização de ações previamente planejadas de contingência; encerramento de trabalhos relacionados a um determinado risco, quando este deixar de existir, entre outras. **Resposta: Controle de riscos.**

CARTA 4

1 – A atividade de Planejar a Gerência de Risco tem a finalidade de definir a estratégia da gestão de riscos, dos recursos necessários para a realização do processo e por fim, da efetivação das ações consideradas necessárias no plano de Gerência de Riscos.

a) Verdadeiro b) Falso

2 – Quais as estratégias de redução de riscos?

- a) Eliminação de riscos, detecção e remoção de riscos e imitação de danos.
- b) Prevenção de riscos, transferência de riscos e imitação de danos.

c) Prevenção de riscos, detecção e remoção de riscos e limitação de danos.

d) Todas as anteriores.

3 – Em qual atividade são caracterizados os aspectos mais importantes de cada risco, com a finalidade de explorar as melhores estratégias de mitigação. De uma forma geral, os riscos são categorizados e

priorizados, segundo algum critério específico estabelecido, para tornar a gerência concentrada nos riscos considerados prioritários. **Resposta: Análise de Riscos**

CARTA 5

1 - Análise dos Riscos é uma série de passos que permitem atacar ativamente os riscos (técnicos e de projeto), como a sua identificação e avaliação, priorização, estratégias de administração dos riscos, monitoramento e sua resolução.

a) Verdadeira b) Falso

2 – Como se classificam os níveis de riscos?

- a) **Intolerável, tão baixos quanto razoavelmente prático, aceitáveis e negligenciáveis.**
- b) Tolerável, tão baixos quanto razoavelmente prático, aceitáveis e negligenciáveis.
- c) Toleráveis, aceitáveis e negligenciáveis,
- d) Nenhuma das anteriores.

3 – Referente a Gerência de Risco na visão CMMI o processo é dividido em três grandes fases, quais são elas? **Resposta: Avaliação de Riscos, Controle de Riscos, Relatórios de Riscos**

APÊNDICE 33 – Pergunta das Cartas - Tema: Manutenção e Evolução de Software

CARTA 1

1 - Entre os problemas comuns na fase de manutenção, podem-se citar a baixa produtividade e o alto custo, além de problemas técnicos, como documentação desatualizada e dificuldade de se alterarem sistemas que foram projetados sem a preocupação com a sua manutenibilidade.

a) Verdadeira b) Falsa

2 - Manutenção é o processo de modificação de um software depois que ele foi colocado em operação. Assinale a alternativa que indica corretamente o tipo de manutenção de software, segundo Pressman, no qual se busca modificar o software a fim de torná-lo mais fácil de serem corrigido, adaptado e melhorado.

- a) Corretiva
- b) Perfectiva
- c) Preventiva**
- d) Adaptativa

3 – Uma atividade ligada às manutenções corretiva e evolutiva de software envolve ter uma boa e

adequada compreensão do sistema a sofrer o processo de evolução, incluindo, por exemplo, uma re-implementação de sistemas legados, de forma a tornar mais simples sua manutenção. Essa atividade recebe a denominação de? **Resposta: reengenharia.**

CARTA 2

1 – Um sistema pode sofrer quatro tipos de manutenção: corretiva, adaptativa ou evolutiva, perfectiva e preventiva.

a) Verdadeiro b) Falso

2 – Sobre as manutenções corretiva e evolutiva de software, é correto afirmar que

- a) a manutenção corretiva visa apenas acertar a documentação do software.
- b) a manutenção corretiva visa, principalmente, reparar erros encontrados no software.**
- c) a manutenção evolutiva não implica alterar o código fonte do software.
- d) a manutenção evolutiva não visa introduzir novas funcionalidades no software.

3 – Um sistema pode sofrer quatro tipos de manutenção, quais são? **Resposta: corretiva, adaptativa ou evolutiva, perfectiva e preventiva.**

CARTA 3

1 - Os sistemas de software estão sujeitos a manutenção, pois, dificilmente, as características que o definem não sofrem modificações durante a sua vida útil. Quanto mais dependentes do mundo real forem os requisitos de um sistema, maior será a probabilidade de esse sistema vir a ser modificado.

a) Verdadeiro b) Falso

2 – São estratégias possíveis de serem adotadas por uma organização para evolução de seus sistemas legados as apresentadas a seguir, **EXCETO**:

- a) descartar o sistema completamente.
- b) substituir todo ou parte do sistema por um novo sistema.
- c) deixar o sistema sem alterações e continuar com a manutenção regular.
- d) realizar engenharia reversa do código do sistema com o objetivo de aumentar sua complexidade ciclomática e torná-lo crítico à empresa.**

3 - Incluir novas funções (ampliações) diz respeito a qual tipo de manutenção? **Resposta: Perfectiva**

CARTA 4

1 – A manutenção preventiva tem como objetivo detectar e corrigir falhas latentes no produto antes que elas virem falhas efetivas.

- a) **Verdadeiro** b) Falso

2 – Quanto à caracterização, a reengenharia de software é classificada como manutenção:

a) **preventiva.**

b) criptográfica.

c) de melhoria.

d) adaptativa.

3 – Quanto à caracterização, a reengenharia de software é classificada como manutenção? **Resposta: preventiva.**

CARTA 5

1 – A manutenção do tipo corretiva modifica o software, para corrigir defeitos identificados pelo cliente.

- a) **Verdadeiro** b) Falso

2 – A respeito de mudança de software, assinale a opção correta.

- a) A manutenção de software é um processo geral de mudanças que ocorre antes ou depois que um sistema é entregue.
- b) **A manutenção preventiva tem como objetivo detectar e corrigir falhas latentes no produto antes que elas virem falhas efetivas.**
- c) A manutenção corretiva para reparar defeitos de software diz respeito à mudança em um sistema para corrigir erros, que podem ser somente de codificação. Erros de projeto e de lógica não estão incluídos nessa categoria.
- d) A manutenção perfectiva tem como objetivo tornar o software perfeito, melhorar seu desempenho e sua manutenibilidade ou adicionar-lhe funcionalidade para satisfazer a novos requisitos.

3 – Nas atividades executadas na manutenção de software, a categoria corretiva se refere à atividade de modificação do software para a correção de erros; a categoria adaptativa se refere a atividades de alteração para adequar o sistema às mudanças no ambiente em que ele esteja operando. Qual é a categoria que contempla as alterações para atender necessidades de evolução requeridas pelo usuário, como mudanças nas regras de negócio.

Resposta: Categoria evolutiva

APÊNDICE 34 – Perguntas das cartas - Tema: Qualidade de Software

CARTA 1

1 – Para se garantir a qualidade dos processos, vários passos devem ser tomados, entre eles: Gerenciar os requisitos, identificando quais são as principais necessidades do software, levando em conta tanto os requisitos funcionais quanto os não funcionais.

- a) Verdadeiro b) Falso

2 – Dentre os atributos de um software de qualidade, incluem-se:

- a) controlabilidade, dependabilidade e eficiência.
- b) controlabilidade, eficiência e manutenibilidade.
- c) eficiência, imutabilidade e manutenibilidade.
- d) **eficiência, manutenibilidade e usabilidade.**

3 – Em relação aos princípios de qualidade, aquele que está diretamente ligado ao tempo de resposta de processamento e aos recursos utilizados no sistema é conhecido como?

Resposta: eficiência

CARTA 2

1 - Acompanhar o projeto de software para que se possa ter uma visão bem realista do progresso do projeto, sendo possível tomar ações eficazes quando o desempenho software se desviar de forma significativa dos planos do projeto, é um passo para se garantir a qualidade dos processos.

- a) Verdadeiro b) Falso

2 - Segundo a ISO/IEC 12119:1994, dentre os requisitos de qualidade de um produto está sua descrição. Um dos objetivos básicos da descrição do produto é o de servir de base para os testes do produto. Dentre os itens que compõem a descrição do produto estão

- a) declarações de usabilidade, declarações de eficiência e declarações de suporte e treinamento.
- b) **identificadores, declarações de funcionalidade e declarações de confiabilidade.**
- c) declarações de manutenibilidade, declarações de portabilidade e declarações de consistência.
- d) declarações de usabilidade, declarações de consistência e indicadores.

3 – O processo de confirmação que um software vai ao encontro das especificações de software se

trata de um conceito- chave de qualidade denominado: **Resposta: Verificação.**

CARTA 3

1 – Gerenciar a configuração do software para estabelecer e manter a integridade dos produtos do projeto ao longo do ciclo de vida do software para dar maior segurança ao desenvolvedor e permitir maior controle de desenvolvimento, é um passo ara se garantir a qualidade dos processos.

- a) **Verdadeiro** b) Falso

2 – Em relação aos princípios de qualidade, aquele que está diretamente ligado ao tempo de resposta de processamento e aos recursos utilizados no sistema é conhecido como:

- a) praticidade;
b) suporte;
c) manutenção;
d) **eficiência;**

3- Responda quais são os Fatores Externos de Qualidade de Software, cite pelo menos 3.

Resposta: Facilidade de uso, Eficiência, Portabilidade, Correção, Robustez e Integridade

CARTA 4

1 – Desenvolver um processo padrão para ser gerenciado e revisado, Identificar os pontos fortes e fracos do processo de desenvolvimento e planejar atividades de melhoramento, é um passo ara se garantir a qualidade dos processos.

- a) **Verdadeiro** b) Falso

2 – O processo de confirmação que um software vai ao encontro das especificações de software se trata de um conceito- chave de qualidade denominado:

- a) Validação.
b) **Verificação.**
c) Precisão.
d) Confiabilidade.

3 – Na prática de garantia de qualidade de software, contrapondo com o controle de qualidade de software, se aplica a atividade de? **Resposta: definição de planos de desenvolvimento de teste.**

CARTA 5

1 – O nível máximo de qualidade de um software é atingido quando os stakeholders estão satisfeitos com os resultados que ele apresenta; para tanto, é essencial que todos os envolvidos no processo de criação desse software façam parte da revisão de qualidade.

A) Verdadeiro **b) Falso**

2 – Qualidade de software é uma área da engenharia de software que tem como objetivo garantir a qualidade pela definição e normatização dos processos de desenvolvimento de sistemas. O grupo de normas técnicas "ISO 9000/2000" define qualidade como o grau em que um conjunto de características inerentes a um produto, processo ou sistema cumpre os requisitos inicialmente estipulados para esses. Assinale a alternativa que melhor define "qualidade", dentro da área de engenharia de software.

a) **Conformidade de um sistema com os requisitos levantados no início do processo de desenvolvimento.**

b) Tempo de vida útil de um sistema e sua efetiva utilidade e aplicação.

c) É medida pelo máximo de tempo de uso entre falhas ocorridas (MTBF) no ciclo de vida do software.

d) Desempenho medido pelo tempo de resposta no processamento e apresentação das informações.

3 – A estratégia de qualidade aplicada à arquitetura tradicional de *software* deve garantir para as etapas de Engenharia de Sistemas, Requisitos e Projetos, respectivamente, os testes de? **Resposta: sistema, validação e integração.**

CARTA 6

1 – O gerenciamento de qualidade de software pode ser estruturado em três atividades principais: garantia de qualidade, planejamento de qualidade e controle de qualidade.

a) Verdadeiro b) Falso

2 – Segundo Pressman, na qualidade do software, as inspeções, revisões e testes utilizados ao longo do processo de software, para garantir que cada produto de trabalho satisfaça os requisitos estabelecidos, são conhecidos como:

a) garantia de qualidade;

b) custo da qualidade;

c) controle de qualidade;

d) reengenharia de processos;

3 – Na avaliação da qualidade de software a medida com que um programa satisfaz sua especificação e cumpre os objetivos visados pelo cliente é chamada de? **Resposta: Corretude**

CARTA 7

1 - O objetivo da atividade de garantia da qualidade é assegurar que os processos e os produtos de software, no ciclo de vida do projeto, estão em conformidade com os padrões, os procedimentos e as descrições de processos definidos para o projeto submetidos a essa atividade.

a) **Verdadeiro** b) Falso

2 – Em relação aos princípios de qualidade em software, no tocante à testabilidade de software, a sentença "quanto menos modificações, menos interrupções no teste" está relacionada com uma característica. Essa característica é identificada como:

a) operacionalidade;

b) compreensibilidade;

c) estabilidade;

d) controlabilidade;

3 – Em relação aos princípios de qualidade em software, no tocante à testabilidade de software, a sentença "quanto menos modificações, menos interrupções no teste" está relacionada com uma característica. Essa característica é identificada como: **resposta: estabilidade.**

CARTA 8

1 - Com relação à qualidade de software, é incorreto afirmar que o processo de validação tem por objetivo realizar testes até que nenhum erro seja encontrado. a) **Verdadeiro** b) Falso

2 – No tocante à garantia de qualidade de software, está relacionada com uma de suas funções:

a) inspeções de requisitos;

b) testes de comparações;

c) auditorias;

d) reengenharia de processos

3– A noção de qualidade de software pode ser descrita por um grupo de fatores, requisitos ou atributos, tais como: confiabilidade, eficiência, facilidade de uso, modularidade, legibilidade, etc; Podemos classificar estes fatores em dois tipos principais, quais são? **Resposta: Fatores externos e Fatores internos.**

CARTA 9

1 – O controle de qualidade de um projeto de software é realizado durante todas as fases do ciclo de vida do projeto.

- a) **Verdadeiro** b) Falso

2 – Na prática de garantia de qualidade de software, contrapondo com o controle de qualidade de software, se aplica a atividade:

- a) executar teste de software
b) desenvolver casos de testes.
c) **definir métricas e medição.**
d) definir estratégias de testes.

3 – Com relação ao conceito de qualidade, a seleção de procedimentos e padrões apropriados adaptados para um projeto de software específico, refere-se a: **Resposta: Planejamento**

CARTA 10

1 – A usabilidade é uma medida de qualidade de software a ser observada também no levantamento de requisitos, que pode ser auferida analisando-se subjetivamente as atitudes dos usuários em relação ao sistema, por exemplo, por meio de um questionário de avaliação.

- a) Verdadeiro b) **Falso**

2 – Com relação à qualidade de software, é **INCORRETO** afirmar que:

- a) os testes são incapazes de garantir que um software não irá apresentar erros durante a execução.
b) os testes são técnicas dinâmicas de verificação e validação, pois trabalham com uma representação executável do sistema.
c) **o processo de validação tem por objetivo realizar testes até que nenhum erro seja encontrado.**
d) um teste bem-sucedido é aquele que revela um erro ainda desconhecido.

3 – O teste de ameaça normalmente deve ser aplicado dentro de um projeto de software nas etapas de: **Reposta: teste de integração e teste de sistema.**

APÊNDICE 35 – Perguntas das cartas – Tema: Testes de Software

CARTA 1

1 – Na disciplina de testes, a implementação de testes, a execução do conjunto de testes e a análise das falhas de testes são atividades de responsabilidade do analista de testes.

a) Verdadeiro b) **falso**

2 – Assinale a alternativa que não corresponde a um dos testes de software, realizados em softwares comerciais.

a) Teste de Sistema

b) Teste de validação.

c) Teste de integração

d) Teste de volatilidade de requisitos

3 – Os testes de software são executados, usando os procedimentos e documentos de script de teste. Para que a fase de execução de teste seja realizada com sucesso deve(m) ser executado(s): **Resposta: os casos de teste**

CARTA 2

1 – Os testes funcionais são caracterizados pelo uso do sistema conforme o seu usuário regular o faria. **a) Verdadeiro** b) falso

2 – A fase de elaboração dos testes de software é uma das partes mais importantes, no desenvolvimento de um software. Sobre o teste de caixa branca, assinale a alternativa correta.

a) Teste feito pelo próprio programador que verifica, se o código que foi construído, é funcional.

b) Teste executado pelo usuário final do software.

c) Teste realizado, na fase de concepção do software.

d) Teste executado, após a implantação do software.

3 – Os testes de software podem ser aplicados no ciclo de desenvolvimento de software através de vários níveis que vão desde o mais elementar até o mais geral. Na etapa de planejamento de testes, inicialmente, a partir do documento de requisitos, é planejado o teste de? **Resposta: Aceitação**

CARTA 3

1 – Os testes de unidade são feitos por equipes especializadas em testes, de forma a se garantir que os

módulos que compõem o sistema sob construção estejam funcionando de acordo com as especificações.

- a) Verdadeiro **b) falso**

2 - Em relação ao teste de software, assinale a alternativa correta sobre o teste de integração:

- a) Tem como objetivo explorar a menor unidade de um projeto

b) Visa testar as falhas decorrentes da integração dos módulos do sistema

- c) Visa descobrir falhas por meio da utilização do mesmo

- d) Teste realizado pelos usuários finais do software.

3 – Tipo de teste que focaliza cada componente de um software de forma individual, garantindo que o componente funciona adequadamente: **Teste de Unidade**

CARTA 4

1 – No teste de integração, verificam-se o funcionamento em conjunto dos componentes do sistema, se são chamados corretamente e se a transferência de dados acontece no tempo correto, por meio de suas interfaces.

- a) **Verdadeiro** b) falso

2 – A atividade de teste é composta por alguns elementos essenciais, que auxiliam na formalização desta atividade. A afirmação serve para selecionar e avaliar casos de teste, de forma a aumentar as possibilidades de provocar falhas ou, quando isso não ocorre, estabelecer um nível elevado de confiança na correção do produto, refere-se a qual elemento da atividade de teste?

- a) Caso de teste.

- b) Procedimento de teste.

- c) Critério de cobertura dos testes.

d) Critério de teste.

3 – Considere: O objetivo é executar o sistema sob o ponto de vista de seu usuário final, varrendo as funcionalidades em busca de falhas em relação aos objetivos originais. Os testes são executados em condições similares às aquelas que um usuário utilizará no seu dia - a - dia de manipulação do sistema. A afirmativa refere-se ao teste de? **Sistema.**

CARTA 5

1 – Em relação a Qualidade e Teste de Software, quando um produto é previamente testado e enviado para uma nova avaliação, Todas as partes alteradas nos documentos, funcionalidades e informações

devem ser testadas como se fosse um produto novo.

- a) **Verdadeiro** b) falso

2 – A atividade de teste é composta por elementos essenciais, que auxiliam na formalização desta atividade. Entre os principais elementos, é correto citar: casos de teste, procedimentos de teste:

a) **critérios de teste e critérios de cobertura de teste**

- b) critérios de teste e casos de uso.
c) casos de uso e diagrama de teste
d) casos de uso e diagrama de classes.

3- No contexto da estratégia para o teste de um projeto, os estágios de teste desempenham um papel importante. O teste que é aplicado a componentes do modelo de implementação para verificar se os fluxos de controle e de dados estão cobertos e funcionam conforme o esperado, é o teste? **Unitário**

APÊNDICE 36 – Perguntas das cartas Hacker – Tema: UML

<p>1 - Para demonstrar elementos estruturais e comportamentais de um sistema, a UML pode utilizar, respectivamente, os diagramas de:</p> <p>a) Atividade e de Sequência.</p> <p><u>b) Pacote e de Atividade.</u></p> <p>c) Sequência e de Objeto.</p> <p>d) Classe e de Pacote.</p>
<p>2 - Os relacionamentos que podem ser representados em um diagrama de caso de uso são:</p> <p>a) associação e polimorfismo, apenas.</p> <p><u>b) dependência, associação e generalização.</u></p> <p>c) associação e generalização, apenas.</p> <p>d) generalização e polimorfismo, apenas.</p>
<p>3 - Enquanto os diagramas de sequência representam cronologicamente a ocorrência dos eventos e as chamadas das mensagens, os diagramas de colaboração representam a organização estrutural dos objetos como eles estão vinculados e as mensagens que trocam entre si. Devido a suas semelhanças, ambos são conhecidos por diagramas de interação.</p> <p>a) <u>Verdadeiro</u> b) Falso</p>
<p>4 - O Diagrama da UML que representa o comportamento interno de determinado objeto, subsistema ou sistema global denomina-se diagrama de:</p> <p><u>a) Estado</u></p> <p>b) Colaboração</p> <p>c) Sequência</p> <p>d) Classe</p>
<p>5 - O diagrama de componentes da UML, que enfoca detalhes de um software, é um tipo especial de diagrama de:</p> <p>a) implementação.</p> <p><u>b) classes.</u></p> <p>c) sequência.</p> <p>d) atividades.</p>
<p>6 - Em um diagrama de classe, o símbolo “+” é chamado de:</p>

<p>a) extensão.</p> <p>b) protegido.</p> <p>c) composição.</p> <p>d) prioritário.</p>
<p>7 - Em UML os diagramas estruturais são organizados em função dos principais grupos de itens encontrados na modelagem de um sistema. Os diagramas estruturais em UML não incluem o diagrama de</p> <p>a) classes.</p> <p>b) caso de uso.</p> <p>c) componentes.</p> <p>d) objetos.</p>
<p>8 - No diagrama de sequência da UML, cada objeto participante é representado por uma caixa e uma linha vertical denominada linha de? Resposta: Vida</p>
<p>9 - No diagrama de classes da UML uma superclasse, com uma ou mais subclasses, representa um relacionamento do tipo:</p> <p>a) composição.</p> <p>b) agregação.</p> <p>c) generalização.</p> <p>d) Associação.</p>
<p>10 - Dentre os diagramas de interação da UML 2.0, linha de vida e foco de controle são duas as características que distinguem os diagramas de:</p> <p>a) sequência dos de comunicação.</p> <p>b) sequência dos de atividade.</p> <p>c) comunicação dos de caso de uso.</p> <p>d) objetos dos de classe.</p>
<p>11 - Considere diversas agências (classe Agencia) de atendimento a reclamações trabalhistas espalhadas em vários pontos do Estado. Uma delas, a central (classe AgenciaCentral), tem atributos diferenciados, porém herda os demais atributos e operações de Agencia. O relacionamento entre essas classes é definido na UML como:</p> <p>a) inclusão.</p> <p>b) composição.</p> <p>c) específico.</p> <p>d) generalização.</p>
<p>12 - Cobre um conjunto de instâncias dos itens encontrados nos diagramas de classe, expressa a parte</p>

estática de uma interação composta pelos objetos que colaboram entre si, mas sem qualquer uma das mensagens passadas entre eles e, também, congela um momento no tempo. Na UML, trata-se do diagrama de:

- a) atividade.
- b) comunicação.
- c) sequência.
- d) **objetos.**

13 - No diagrama de casos de uso, os relacionamentos podem ser associações ou generalizações, podendo ocorrer entre atores e, nos casos de uso, são somente generalizações.

- a) Verdadeiro
- b) **Falso**

14- Os diagramas da UML 2.0 são divididos em três categorias de diagramas:

- a) estruturais, temporais e de instalação.
- b) comportamentais, temporais e de comunicação.
- c) **comportamentais, estruturais e de interação.**
- d) comportamentais, temporais e de atividades.
- e) estruturais, comportamentais e de instalação.

15 - O conceito de ator pode ser representado graficamente na UML 2.0 nos diagramas de

- a) sequência e de componentes.
- b) **sequência e de casos de uso.**
- c) casos de uso e de tempo.
- d) casos de uso e de componentes.

16 - Diagramas de interação são utilizados na UML para modelagem dos aspectos dinâmicos do sistema. No diagrama de sequência um diagrama de interação em que é dada ênfase à ordenação temporal das mensagens, é explicitamente representada a linha de vida do objeto, bem como o período durante o qual ele está desempenhando uma ação.

- b) **Verdadeiro**
- b) Falso

17- A respeito dos diagramas da UML, os diagramas em UML podem ser estáticos ou dinâmicos. O diagrama de classes é um exemplo de um diagrama dinâmico.

Verdadeiro

- b) **Falso**

18 - Um diagrama de estados da UML especifica os estados possíveis, quais transições são permitidas entre os estados, quais eventos ocasionam as transições e qual comportamento é executado em resposta aos eventos.

<p>a) <u>Verdadeiro</u> b) Falso</p>
<p>19 - Os elementos de um diagrama de atividades podem ser divididos em dois grupos: os que são utilizados para representar fluxos de controle concorrentes e os que são utilizados para representar controles paralelos.</p> <p>Verdadeiro <u>b) Falso</u></p>
<p>20 - O diagrama de implantação da UML representa a topologia física de um sistema de software e, opcionalmente, os componentes que são executados nessa topologia, cujos elementos são nós e conexões.</p> <p>a) <u>Verdadeiro</u> b) Falso</p>

APÊNDICE 37 – Perguntas das cartas Hacker – Tema: SORTE

<p>“Tente uma, duas, três vezes e se possível tente a quarta, a quinta e quantas vezes for necessário. Só não desista nas primeiras tentativas, a persistência é amiga da conquista. Se você quer chegar a onde a maioria não chega, faça o que a maioria não faz.” Bill Gates</p> <p>PERSISTA, NÃO DESISTA! AVANCE 5 CASAS!</p>
<p>“Não deixe o ruído das opiniões de outras pessoas calar sua própria voz interior. Tenha coragem de seguir seu coração e intuição.” Steve Jobs</p> <p>SIGA EM FRENTE! AVANCE 5 CASAS!</p>
<p>“Você pode encarar um erro como uma besteira a ser esquecida, ou um resultado que aponta uma nova direção”. Steve Jobs</p> <p>NÃO TENHA MEDO DE ERRAR! ARRISQUE-SE! AVANCE 5 CASAS!</p>
<p>“A melhor maneira de prever o futuro é inventá-lo.” Alan Kay</p> <p>REINVENTE-SE! AVANCE 5 CASAS!</p>
<p>“Os que são loucos o suficiente para pensarem que podem mudar o mundo, são os que o fazem.” Steve Jobs</p> <p>FAÇA A DIFERENÇA! AVANCE 5 CASAS!</p>

APÊNDICE 38 – Componentes principais do jogo SoftGame.



Fonte: Autor (2016)

APÊNDICE 39 – Manual e acessórios do jogo SoftGame



Fonte: Autor (2016)

APÊNDICE 40 – Todos Componentes do jogo SoftGame



Fonte: Autor (2016)